

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-265379

(43)公開日 平成9年(1997)10月7日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/16	3 2 0		G 0 6 F 3/16	3 2 0 B
				3 2 0 A
	3/14	3 1 0		3 1 0 B
G 0 6 T 1/00			G 0 9 G 5/00	5 1 0 H
G 0 9 G 5/00	5 1 0		G 1 0 L 3/00	5 5 1 G

審査請求 未請求 請求項の数15 O L (全 36 頁) 最終頁に続く

(21)出願番号 特願平9-26968

(22)出願日 平成9年(1997)2月10日

(31)優先権主張番号 9 6 0 2 7 0 1 6

(32)優先日 1996年2月9日

(33)優先権主張国 イギリス (G B)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 エリ ツィルケル-ハンコック

イギリス国 ジーユー4 7エイチエイチ

スアレイ、ギルドフォード グリーン

ヒルガーデنز 50

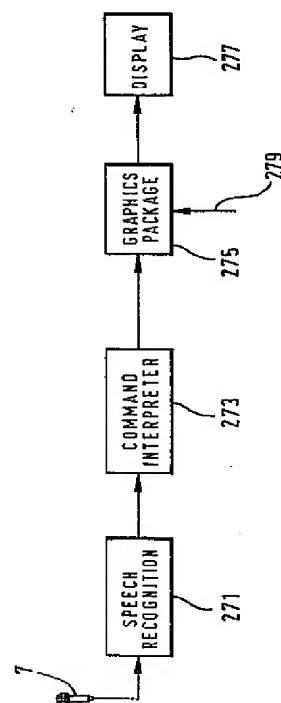
(74)代理人 弁理士 大塚 康徳 (外1名)

(54)【発明の名称】 画像操作装置及び方法

(57)【要約】

【課題】 色編集用語または色操作ツールを用いることなく、画像を編集できる画像操作装置および方法を提供する。

【解決手段】 画像の色を操作する装置は、ユーザコマンドを示す電気音声信号を供給するマイクロホン7、入力音声信号を認識する音声認識装置271、認識音声を解釈するコマンドインタプリタ273、コマンドインタプリタに応答するグラフィックパッケージ275および編集されるべき画像を表示するディスプレイ277を備える。本装置は、例えばポインティング装置からの他の入力279を受け入れる。



**【特許請求の範囲】**

【請求項1】 画像を再生する画像再生手段と、前記再生画像に対する所望の色変化を表わす表現を含む音声信号の入力を可能とする音声認識ユーザインタフェースと、前記音声認識インタフェースから出力された認識結果を解釈する解釈手段と、前記解釈手段にตอบสนองして、ユーザが必要とするカラー操作を行うべく、前記再生画像の1つ以上の部分の色を変化させる変化手段とを備えることを特徴とする画像操作装置。

【請求項2】 前記画像再生手段は表示装置を備え、当該装置は、変化されるべき表示画像の1つ以上の領域を指定する指定手段を更に備えることを特徴とする請求項1に記載の画像操作装置。

【請求項3】 請求項2記載の装置において：前記指定手段は、変化されるべき領域を移動するポインティングデバイスであり、前記変化手段は、前記ポインティングデバイスによってたどられる画素の色相値の統計的特徴を決定し、決定された統計的特徴と類似する色相値を有する画像のすべての画素の色相値を変化させることを特徴とする請求項2に記載の画像操作装置。

【請求項4】 前記変化手段は、前記ポインティングデバイスによってたどられた画素の平均値および標準偏差値を決定し、該決定された平均値を中心とした該決定された標準偏差の範囲内にある色相値を有する当該画像内のすべての画素を、前記変化手段によって変化させることを特徴とする請求項3に画像操作装置。

【請求項5】 色相変化の量が、所望の色変化の色相値に依存することを特徴とする請求項3または4記載の画像操作装置。

【請求項6】 前記画像再生手段は、印刷装置を備えることを特徴とする請求項1乃至5のいずれかに記載の画像操作装置。

【請求項7】 請求項1乃至6のいずれかに記載の画像操作装置を有することを特徴とするコンピュータ。

【請求項8】 請求項1乃至6のいずれかに記載の画像操作装置を有することを特徴とするファクシミリ装置。

【請求項9】 請求項1乃至6のいずれかに記載の画像操作装置を有することを特徴とする複写装置。

【請求項10】 請求項1乃至6のいずれかに記載の画像操作装置を有することを特徴とするカメラ。

【請求項11】 音声認識ユーザインタフェースを用いて、ユーザによって発声入力された所望の色変化に関する文に対する認識結果を出力し、出力された認識結果を解釈し、画像の全体もしくは部分の色を変化させ、ユーザが所望する色操作を行うことを特徴とする画像操作方法。

【請求項12】 前記画像が画像を操作する前に再生されることを特徴とする請求項11に記載の画像操作方

法。

【請求項13】 請求項11または12に記載の方法を実行するソフトウェアがプログラムされたデータ搬送媒体。

【請求項14】 添付図面の図36から図38を参照して記述されるように、音声コマンドによって実質的な画像の色操作を行う画像操作方法。

【請求項15】 添付図面の図36から図38を参照して記述されるように、画像の色操作を行う画像操作装置。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】本発明は、画像を操作する方法および装置に関する。特に、本発明は、コンピュータグラフィックシステムにおける画像の色操作に好適なものである。

**【0002】**

【従来の技術】今日のコンピュータグラフィックシステムは複雑で、画像の色を操作する多くの処理オプションをユーザに提供する。このようなシステムにおいて、ユーザは、画像の色を記述するために用いられる、例えば、色相、明度および彩度等の専門用語を十分理解し、所望の画像の色操作をするために提供された色操作ツールに精通していなければならない。

**【0003】**

【発明が解決しようとする課題】EP594129（出願人：日立製作所）には、ポインティングデバイスとともに音声によって制御されるグラフィックパッケージが開示されている。画像内のオブジェクトを移動させたり、画像内でパンやズームを行う処理が想定されている。しかしながら、EP594129には、画像の一部または複数部分の色を操作する場合に生じる困難な問題について記載されていない。

【0004】本発明は上記従来技術に鑑みてなされたものであり、色操作を表現する音声入力によって画像操作を行うことを可能とし、容易に所望の画像操作を行える画像操作装置及び方法を提供することを目的とする。

**【0005】**

【課題を解決するための手段】本発明の一態様による画像操作装置は、画像を再生する手段と、ユーザが再生画像に行われた所望の色変更の内容表現を有する音声信号を入力可能とする音声認識ユーザインタフェースと、ユーザが必要とする色操作を行うため一つまたはそれ以上の部分の再生画像の色を変更する手段とを備える。

【0006】また、変化させるべき画像の部分は、変化させるべき領域内の画素の色相値のサンプルを取るために用いられるポインティングデバイスによって指示されてもよい。

【0007】本発明の別の態様による画像操作装置は、操作されるべき画像を再生し、音声認識ユーザインタフェースを用いてユーザが発声入力した所望の色変化の内

容表現に対する認識結果を出力し、認識結果出力を解釈してユーザが必要とする色操作を遂行すべく再生画像を全体的にまたは部分的に変化させる。

【0008】

【発明の実施の形態】以下、添付の図面を参照して本発明の一実施形態を説明する。

【0009】本発明の実施形態はコンピュータのハードウェアで実施され得るが、以下に説明する実施形態は、例えばパーソナルコンピュータ、ワークステーション、複写機、ファクシミリ等の装置の処理ハードウェアとともに動作するソフトウェアにおいても用いられ得るものである。

【0010】図1は、本発明の一実施形態を処理するように構成されたパーソナルコンピュータ（PC）1を示す。キーボード3、ポインティングデバイス5、マイクロフォン7および電話線9は、インタフェース11を介して接続される。キーボード3およびポインティングデバイス5によって、ユーザはシステムを制御できる。マイクロフォン7は、ユーザの音響的音声信号をそれに相当する電気信号に変換し、この電気信号をPC1に処理のために供給する。本実施形態において、処理されるべき入力音声の始点および終点は、入力発声音が継続されている間キーボード3上のスペースバーを押すことによって認識される。このような方法で、システムは認識されるべき入力発声音を処理する。内蔵モデムおよび音声受信回路（図示されていない）は電話線9に接続され、PC1は、例えば遠隔地のコンピュータまたはユーザと通信することができる。

【0011】本発明で規定するようにPC1を動作させるためのプログラム命令は、既存のPCで用いるために磁気ディスク13のような記憶装置上に供給されてもよいし、または電話線9を介してリモートコンピュータと通信する内蔵モデムによって供給されてもよい。

【0012】本実施形態の限定された語彙による連続音声認識システムの処理を、図2を参照して説明する。例えば、マイクロフォン7からの入力音声を示す電気信号は、プリプロセッサ（PREPROCESSOR）15に入力される。プリプロセッサ15は、入力音声信号をパラメータフレームのシーケンスに変換する。ここで、各パラメータフレームは、入力音声信号に対応する時間フレームを示すものである。次に、パラメータフレームのシーケンスは、認識ブロック（RECOGNITION）17に供給される。認識ブロック17において、入力されたパラメータフレームのシーケンスを標準モデル即ちワードモデル（WORD MODELS）19と比較することによって、入力音声は認識される。標準モデル、即ちワードモデル19の各モデルは、認識されるべき入力音声と同様の種類のパラメータで表されるパラメータフレームのシーケンスで構成されている。

【0013】言語モデル（LANGUAGE MODEL）21および

ノイズモデル（NOISE MODEL）23も認識ブロック17に対する入力として供給され、認識プロセスにおいて用いられる。本実施形態において、ノイズモデルは沈黙またはバックグラウンドのノイズを示し、認識されるべき入力音声信号のパラメータフレームと同様のタイプの1つのパラメータフレームから構成される。言語モデル21が用いられ、システムに認知されたワードのシーケンスに適應させるように、認識ブロック17から出力されたワードの取り得るシーケンスを制限する。認識ブロック17から出力されたワードシーケンスはコピーされ、例えばワードプロセッシングパッケージ等で用いられる。または、そのワードシーケンスは、オペレータコマンドとして用いられ、PC1の動作を開始したり、中止させたりまたは修正したりする。

【0014】以下、上記の装置に関するより詳細な説明を行う。

【0015】＜プリプロセッサ＞プリプロセッサ15について図3～図10を参照して説明する。

【0016】プリプロセッサ15の機能は、必要な情報を音声から抽出することにより処理されるべきデータを削減することである。音声分析の技術分野において、音声を前処理する方法は多数知られており、以下に説明する方法は、一実施形態として示されているものである。よって、以下に説明するような方法に限定されると解釈されるべきではない。本実施形態において、プリプロセッサ15は、「フォルマント（formant）」関連情報を抽出するように設計されている。フォルマントとは、ユーザの声道の共振周波数として定義され、声道の変化にしたがって変化する。

【0017】図3は、入力音声信号に対して行われる前処理を示すブロック図である。マイクロフォン7または電話線9からの入力音声S(t)は、フィルタブロック（FILTER）61に供給され、入力音声信号内のあまり重要でない情報を含む周波数を除去する。音声信号の中で、もっとも重要な情報は、4KHz以下の周波数に含まれる。したがって、フィルタブロック61は、4KHzを上回るすべての周波数を除去する。その後、濾過音声信号は、ADコンバータ（ADC）63によって、デジタルサンプルに変換される。ナイキストサンプリング基準にしたがって、ADC63は、8000回/秒のレートで濾過信号をサンプリングする。本実施形態において、全入力発声音はデジタルサンプルに変換され、音声信号を処理するための次のステップの前に、バッファ（図示されていない）に記憶される。

【0018】入力音声は抽出された後、入力音声は、分割ブロック（DIVISION INTO FLAMES）65で互いに一部が重なり合う等しい長さのフレームに分割される。入力音声をフレームに分割する理由をより詳細に説明する。上記のとおり、連続的に音声が発せられる間、フォルマント関連情報は連続的に変化し、その変化の割合は、生

理学的抑制によって制限される音声発声器官の動作の割合に直接的に関連する。したがって、変化するフォルマント周波数を追跡するためには、音声信号は短時間の周期または短いフレームで分析されなければならない。この方法は、音声分析の技術分野において、「短時間」音声分析として知られている。短時間分析を行う際、考慮しなければならない点が2つある。すなわち、(i)どのくらいの速度で、時間フレームが音声信号から抽出されなければならないか、(ii)どのくらいの大きさの時間フレームが用いられなければならないか、という2点である。

【0019】第1の考慮に関しては、音声発声器官の動作速度に依存する。すなわち、重要なイベントを見逃さないことを保証するとともに、適当な連続性を保証するように、フレームは十分に近寄っている必要がある。本実施形態において、1つのフレームは10msecに1回の割合で抽出される。第2の考慮に関しては、フレーム間の音声信号の特性が一定になるような十分に短い時間フレームと、フォルマントを区別するのに十分な周波数を提供するのに十分に長い時間フレームとの間の妥協しうる点で決定される。本実施形態において、フレームの長さは20msecであり、上記サンプリング速度で、1つのフレームにつき160個のサンプルに相当する。

【0020】これらのフレームを音声サンプルから直接抽出することによって発生させる場合、かなりの周波数のひずみが生じる。したがって、このようなひずみを低減するため、スムーズウィンドウ機能を用いるべきである。ハミング(Hamming)、ハニング(Hanning)、ブラックマン(Blackman)、バートレット(Bartlett)およびカイザー(Kaiser)等のようなウィンドウは多数存在し、何れも適用可能である。これらすべてのウィンドウは、音声分析の当業者に知られている。本実施形態において、ハミングウィンドウが用いられ、これは以下の等式によって表される。

【0021】

【数1】

$$W(n) = 0.54 - 0.46 \cos \left[ \frac{2\pi n}{N_s - 1} \right] \quad \dots (1)$$

【0022】ここで、 $N_s$ はウィンドウ内のサンプルの数、すなわち、160個のサンプルを示す。

【0023】図4は、本実施形態において行われる短時間分析処理を詳細に示す図である。フレーム1(Frame 1)、すなわち、時刻aと時刻bとの間の音声信号に、上記の式(1)で求められるウィンドウ関数が乗算される。さらに、選択されたフレームの速度およびフレームの長さによって、次のフレームであるフレーム2は、フレーム1の中間、すなわち時刻cから始まる。

【0024】入力音声信号のフレームがいったん抽出されると、ブロック67(DFT67)においてフレーム

の離散フーリエ変換(DFT)の振幅が計算される。すなわち、 $|S_k(f)|$ が計算される。ここで、 $f$ は離散周波数変数を示す。このプリプロセッサの多くの態様は、人間の聴覚システムの動作をシミュレートするために設計されているので、振幅情報のみが必要であり、入力音声信号の位相に対しては比較的敏感でない。図5は、160個のサンプル、すなわち $r=0, 1, \dots, 159$ から構成される1つのフレーム $S_k(r)$ の典型的な音声信号を示す。高速フーリエ変換(FFT)アルゴリズムをDFTの計算で有効に用いるため、フレーム $S_k(r)$ 内のサンプルの数を2のべき乗まで増加させる必要がある。これを達成する方法の1つとして、160個のサンプルの後に96個の0を追加し、256個のサンプルを得ることが挙げられる。この技術は「0の充填」として知られ、音声分析の分野において公知であるので、これ以上の説明はしない。

【0025】 $S_k(r)$ のDFTをコンピュータで演算する場合、計算処理のために、スペクトルの第1の128個のサンプルだけが必要である。なぜならば、音声は実際の信号であり、後半の128個のサンプルは前半の128個のサンプルの鏡像だからである。図6は、図5に示されるフレーム $S_k(r)$ 内の音声信号のDFT $|S_k(f)|$ の振幅の最初の128個のサンプルを示し、最終のサンプルはサンプリング周波数の半分の周波数、すなわち、4KHzの周波数で発生するものとなる。

【0026】上記のとおり、プリプロセッサ15の目的は、データレートを低減し、入力音声信号の特定成分を強調することである。1つのフレームにつき128個のサンプルしか存在しないので、DFTによるデータレートの低減はわずかである。さらにデータレートを低減する方法の1つとして、スペクトルを等しい周波数帯に分割し、各周波数帯内のサンプルを平均化することが挙げられる。すなわち、図6に示されるサンプルをフィルタバンクに通過させることである。

【0027】人間の聴覚システムに関する研究によれば、耳の周波数分解能が周波数が増加するとともに減少することが示されている。従って、高周波数領域と比較して低周波数領域の中に多くの周波数帯を有するフィルタバンク、すなわち対数スペースフィルタバンクは、直線スペースフィルタバンクよりも好ましい。なぜなら、対数スペースフィルタバンクの方が知覚的に重要な情報をより多く保持することができるからである。

【0028】本実施形態において、40個の周波数帯を有するメルスペースフィルタバンク(MEL FILTER BANK)69が用いられる。メルスケールは音声分析の分野で公知であり、音声の知覚周波数を線形スケール上にマップしようとする対数スケールである。図7は、メルスペースフィルタバンク69の出力 $|S_k(f')|$ を示し、図6に示されるサンプルがバンク69を通過したときの出力を示す。振幅スペクトルの結果エンベロープ100

は、フィルタバンク69の平均化効果のためにかなりなめらかであるが、フィルタバンクの対数スペースのために低周波数帯ではなめらかではない。

【0029】次に、フォルマント関連情報が、図3のブロック71、73および75を用いて、下記のプロセスによって音声から抽出される。

【0030】励起信号 $E(t)$ およびフィルタ $V(t)$ の項を用いて、ユーザの音声信号 $S(t)$ をモデルにすることができる。ここで、励起信号 $E(t)$ は声道に入る空気流を示し、フィルタ $V(t)$ は声道の共振効果を示す。結果的に、音声信号の周波数スペクトル $|S(f)|$ の振幅は、励起信号の周波数スペクトル $|E(f)|$ の振幅と声道フィルタのスペクトル $|V(f)|$ の振幅を乗算することによって求められる。すなわち、以下の(2)式によって求められる。

【0031】

【数2】

$$|S(f)| = |E(f)| \cdot |V(f)| \quad \dots(2)$$

【0032】ケプストラム法(cepstral method)として知られる方法、すなわち、音域情報を入力音声から抽出する方法の一つを以下に示す。この方法は、音声振幅応答 $|S(f)|$ の対数を取ることによって、音域フィルタ振幅応答 $|V(f)|$ を励起振幅応答 $|E(f)|$ から分離することを含む。この結果、励起および声道フィルタの特性の和で表わされるようになる。すなわち、(3)式によって求められる。

【0033】

【数3】

$$\log|S(f)| = \log|E(f)| + \log|V(f)| \quad \dots(3)$$

【0034】図8は、メルフィルタバンク69からのログ出力のエンベロープ、すなわち、 $\log|S(f)|$ を示し、2つの成分101および103の加算性を図示するものである。成分101は、声道特性、すなわち、 $\log|V(f)|$ を示す。成分103は、励起特性、すなわち、 $\log|E(f)|$ を示す。成分101のピークは、声道のフォルマント周波数で発生する。また、成分103の等間隔のピークは、話者のピッチのハーモニク周波数で発生する。

【0035】声道特性101は、ブロック(LOG)71から出力されたサンプルに離散コサイン変換(DCT)を行い、その結果をフィルタリングすることによって、励起特性103から抽出できる。

【0036】図9は、DCTブロック73の出力を示す。これは、ケプストラム(cepstrum) $C_k(m)$ として知られている。ケプストラムの独立変数(図9のx軸)は時間の次元を有し、「クエフレンシ(quefrensy)」と名づけられている。図8に示される強い周期性を有する成分103は、話者のピッチ周期 $T$ に等しい位置で、ケプストラムのピーク105になる。また、図8に示され

るゆっくり変化する成分101は、ケプストラムの原点付近の多数の小さいピーク10に変換され、その位置および振幅はフォルマントに依存する。

【0037】音声の声道特性および励起特性がクエフレンシスケール上の分離した部分にあらわれるので、これらは、共振プロセスまたはケプストラム用語で言ういわゆる「リフタリング(liftering)」プロセスによって、互いに分離される。図9に示されるケプストラム $C_k(m)$ は、1組の離散ケプストラム係数(discrete cepstral coefficient)( $C_0, C_1, \dots, C_{39}$ )から構成される。したがって、リフタリングは、1つの矩形ウィンドウによって達成される。しかしながら、信頼性が劣ると考えられるスペクトルの部分を強調しないようにするため、より漸次的なウィンドウ機能が好ましい。本実施形態において、以下のウィンドウ機能が、リフタリングブロック(LIFTERING)75で用いられる。

【0038】

【数4】

$$W_{lf}(m) = 1 + \frac{N_c}{2} \sin \frac{\pi m}{N_c} \quad \dots(4)$$

【0039】ここで、 $N_c$ は、リフタリングブロック75から各フレーム毎に出力されたケプストラム係数の所望の個数であり、本実施形態では12である。

【0040】また、上記の12個のケプストラム係数に加えて、各フレーム内の音声信号のパワー、すなわち、「フレームパワー」もまた計算される。これは、フレーム内の入力音声信号が発生音声信号に対応するか否かを示すのに用いられるので、とりわけ重要な特徴である。フレームパワーは、音声分析の分野では公知の一般的な方法を用いて、図3に示されるフレームパワーブロック(FRAME POWER)81で計算される。変化しうる記録環境や音の大きさ等に影響されないように、ブロック81で決定されたパワーは、パワー正規化ブロック(POWER NORMALIZING)83で正規化され、パワー係数 $P_k$ を求める。パワー係数 $P_k$ は、リフタリングブロック75から出力されたケプストラム係数に結合される。パワーは、バッファ(図示されていない)に記憶された発声音全域にわたって最大パワー(dB)を決定し、これを各フレームのパワーから減じ、その結果と正規化定数とを掛けることによって、ブロック83で正規化される。

【0041】本実施形態においては、各フレーム内の入力音声信号のパワーが決定されるが、各フレーム内の入力音声信号を示す他の値も用いられる。例えば、フレーム内の入力音声信号の平均振幅の尺度が決定され、正規化され得る。

【0042】以上のように、プリプロセッサ15は、各時間フレームで、1組の係数、すなわち、12個のケプストラム係数および1つのパワー係数を出力する。便宜上、フレーム $k$ を示す係数は、パラメータフレーム $f_k$ として示され、次のフレームを示す係数は、パラメータ

フレームfk+1等として示す。

【0043】＜バッファ＞プリプロセッサ15から出力されたパラメータフレームfkは、図2に示されるバッファ（BUFFER）16に供給される。本実施形態において、バッファ16は、ADC63から出力された入力音声のデジタルサンプルを記憶するバッファ（図示されていない）に記憶された音声に関して生成されたすべてのパラメータフレームを記憶するのに十分な容量を有する。全入力発声音がプリプロセッサ15によって処理された後、バッファ16に記憶されたパラメータフレームは、それらが受信された順番で認識ブロック17に供給され、入力発声音が認識される。

【0044】＜標準モデル＞上記のとおり、どのワードがプリプロセッサ15からの出力信号によって表現されているかを決定するため、出力信号は記憶されている標準モデルと比較される。この記憶された標準モデルは、システムに既に認知されたワードおよびシステム周辺の音響環境をモデルにしたものである。特定のワードに関連する各モデルは、上記プリプロセッサ15から出力されたパラメータフレームと同じタイプのパラメータフレームのシーケンスで構成される。しかしながら、ワードモデルのフレームと認識されるべき入力発声音のフレームとを区別するため、ワードモデルのフレームはステート（state）と表現することにする。

【0045】本実施形態による音声認識システムの1つの特徴は、ワードモデル、環境（または、ノイズ）モデルまたは言語モデルがシステムに前もって記憶されていないエンドユーザに供給され得ることである。これによって、ユーザは自由にシステムを学習させ、ユーザが必要とするフレーズを認識させるようにすることができる。さらに、ユーザにとって有用でないワードが予め記憶されていることで、システムが過負荷となることを防止できる。さらに、以下の記載から明らかなように、記載された特定の学習方法はこの状況において特に適している。なぜならば、それは、学習のために費やされる時間を必要とせず新しいフレーズがシステムによって学習されることを可能とするからである。さらに、標準モデルは、すべてのワードに対応し、音素には対応しないので、システムは、いかなる言語または言語の組み合わせに対しても動作する。学習プロセスを、図10から図17を参照してより詳細に説明する。

【0046】＜学習＞図10は、学習プロセスで用いられる構築／更新モジュール（BUILD/UPDATE MODULE）91を示す概略図である。特に、構築／更新モジュール91は、システムによって学習されるべき1つ以上のワードを示すパラメータフレームfkのシーケンスと、矢印92で示される1つ以上の入力音声ワードに対応するテキストを示すユーザ情報とを受ける。十分な情報が構築／更新モジュール91に入力されると、入力ワードに対応するワードモデルを発生し、言語モデル21を更新す

る。本実施形態において、ワードモデルおよび言語モデルの両方は、ハードディスク93のような大容量のデータ記憶装置に記憶される。

【0047】ノイズモデル23が本実施形態において決定される方法を以下に説明する。まず、ユーザは新しいノイズモデルを構築したり、または既存のノイズモデル23を変化させる必要があることを指示する。この指示に応じて、システムは、ユーザに沈黙を示す音を入力するよう促す。この処理は、沈黙状態の間、ユーザがキーボード3上のスペースバーを押すことによって行われる。沈黙の周期の終わりには、ユーザはワードを発生しなければならない。システムが沈黙を示すべく生成されたパラメータフレームのパワー係数を正規化できるようにするためである。ユーザが沈黙の周期の終わりにワードを発声しない場合、ノイズモデル23のパワー係数が非現実的に高くなり、非認識エラーが生じることになる。最後に、ノイズモデル23を決定するため、システムは沈黙の期間に対して生成されたパラメータフレームを平均化し、ノイズモデル23として用いられる1つのパラメータフレームを形成する。

【0048】本実施形態の音声認識システムは、連続的に発声されるワード、すなわち、フレーズ内に含まれるワードを認識するために設計されている。良い認識結果を得るため、標準モデル（または、連続ワードモデル）は、重要なワードを含むフレーズ例から得るべきである。好ましくないことに、連続的に発声されたフレーズ内のワードの始めと終わりを確認する作業は容易ではない。本実施形態が連続ワードモデルを発生させる方法の概略を以下に示す。まず、システムは、分離的に発声されたワードからワードモデルを決定する。このモデルは、分離ワードモデルとして示す。しかし、音声認識の当業者が以下の記載から理解するように、これらの分離ワードモデルは、本技術分野における一般的な分離モデルには相当しない。システムは分離ワードモデルを用い、分離ワードモデルとそれに相当するワードを含むフレーズ例とを比較することによって、連続ワードモデルを生成する。

【0049】分離ワードモデルを生成するため、ワードは、マイクロフォン7または電話線を介して分離的にシステムに入力されなければならない。上記のとおり、スペースバーが用いられ、各入力発声音を確認する。したがって、分離的に発声されたワードの発声音を示すパラメータフレームのシーケンスは、沈黙に対応するパラメータフレームをワードの始めと終わりに有する。次にシステムは、分離ワードの発声音と当該ワードを含むフレーズ例とを比較する。この比較によって、分離的発声音内のワードのおよその始めと終わりを確認する。この始点と終点は平均化され、そのワードに対する分離ワードモデルは、確認された始点と終点との間にある平均化されたパラメータフレームのシーケンスを抽出することに

よって、決定される。この方法で分離ワードモデルを決定することによって、ワードの始めと終わりの沈黙を除去するだけではなく、そのワードの部分であって、連続的な発声音声の間において発音されない部分も除去される。したがって、本実施形態の分離ワードモデルは一般的な分離的ワードモデルには対応せず、沈黙を入力発声音の始めと終わりから除去することによって決定され、連続音声が発せられているときのワードをより適切に表すものとなる。

【0050】いったん分離ワードモデルが決定されると、フレーズ内の当該ワードの位置を確認するために、当該ワードを含むフレーズ例と整合される。最後に、標準ワードモデルまたは連続ワードモデルは、フレーズにおける確認位置からの音声を抽出及び結合することによって、決定される。システムがワードモデルを生成する方法の詳細について、以下に説明する。

【0051】ユーザがシステムに1つ以上の新しいフレーズを教えたい場合、図11に示されるサブルーチンを起動する。ステップS1で、ユーザは、新しいワードまたはフレーズのテキストをキーボード3を介してシステムに入力する。システムは、ステップS3において、そのワードまたはフレーズが既知のものであるか否かをチェックし、そのワードまたはフレーズが未知のものであれば同じワードまたはフレーズをマイクロフォン7を介して入力するようユーザに促す。そして、ステップS1に入力されたワードまたはフレーズに対応するテキストに、マイクロフォン7を介して入力されたフレーズを関連させる。次にステップS5において、PC1は、前もって入力されていたすべてのテキストを用い、フレーズ内のどのワードが（分離された状態で）既知となっているかをチェックし、未入力（未知）のワードをマイクロフォン7を介して入力するようユーザに促す。

【0052】ステップS5の後、ユーザは、ステップS7にて他の新しいワードまたはフレーズを入力するかどうかを決定する。新たに入力する場合は、ステップS1に戻る。他方、ユーザがこれ以上ワードを入力しないことを決定した場合、処理はステップS9に移動する。ステップS9において、標準モデルは、システムに入力された少なくとも2つのフレーズに含まれる未知ワードについて生成される。例えば、学習フレーズが入力されておらず、システムには標準モデルがまだ記憶されていない状態で、ユーザが「get an image（画像を得る）」および「get the earth（接地する）」という2つのフレーズを入力し、ステップS7でユーザがこれ以上フレーズを入力する意志がないことを指示した場合を説明する。この場合、入力フレーズの両方に含まれるワードは「get（得る）」の1つだけなので、システムは「get」というワードのワードモデルを生成するだけでよい。他方、ユーザが「get an image（画像を得る）」というフレーズを2回入力する場合、システムは、当該フレーズ内の

各ワードの標準モデルを生成させ得る。さらに、上記の前者の例を取れば、ユーザが「get an image（画像を得る）」および「get the earth（接地する）」という2つのフレーズを入力した後、第2の学習セッションで「get the earth（接地する）」というフレーズを入力した場合、「get（得る）」というワードは既に分離して入力されているので、システムは、ステップS5で「get（得る）」というワードを入力する必要がないことをユーザに伝える。さらに、「the（その）」及び「アース（earth）」というワードはシステムに入力されている2つのフレーズによって表されるので、システムは「the（その）」及び「アース（earth）」というワードの標準モデルを発生させることができる。この方法で学習を進めることにより、ユーザはシステムをユーザにとって都合よく学習させることができる。

【0053】図12を参照すると、上記方法で入力された各フレーズは、フレーズ番号Pを付され、ハードディスク93のフレーズアレー115に記憶される。同様に、入力される各分離ワードにはワード番号Wが付され、ハードディスク93のワードアレー117に記憶される。図12に示されるように、フレーズアレー115内の各フレーズPは、自身に関連する次のようなデータを有する。すなわち、パラメータフレーム（PARAMETER FRAMES）123、フレーズを形成するワードシーケンス（PHRASE WORDS）125およびステータスフラグ（STATUS）127を有する。ワードシーケンス125内の各ワードは、自身に関連するテキスト（TEXT）129、フレーズ番号P、ワード番号（WORD No.）131（すなわち、フレーズ内のワードの位置）、フレーズ内のワードの（最初は認知されていない）時間境界（TIME BOUNDRIES）133、およびワードアレー117内の対応するワードWを示すワードインデックス（WORD INDEX）135を有する。

【0054】ワードアレー117内の各分離ワードWは、自身に関連する次のようなデータを有する。すなわち、パラメータフレーム（PARAMETER FRAMES）137、当該ワードが見出されたフレーズへの戻りを示す発声音インデックス（UTTERANCE INDEX）139およびステータスフラグ（STATUS）141を有する。まず、分離ワードおよび分離フレーズがシステムに入力されると、各ワードまたはフレーズに関連するステータスフラグ127および141は、それらがまだ処理されていないことを示すFLEXIにセットされる。なお、時間境界133は、フレーズのワードシーケンス内の各ワード毎に存在し、UNKNOWN（未知）に設定される。

【0055】次に、図11のステップS9で行われる未知ワードのワードモデルを生成するための処理の詳細を図12から図16を参照して説明する。ここでは、学習フレーズとして「get an image（画像を得る）」というフレーズが2回、「get the earth（接地する）」という



フレーズが1回入力された場合を例として用いて説明する。したがって、フレーズアレー115には3つの要素P1、P2およびP3が存在し、「get an image（画像を得る）」という発声音のフレーズが2つと「get the earth（接地する）」という発声音のフレーズが1つ含まれる。さらに、ワードアレー117には5つの要素W1、W2、W3、W4、W5があり、上記2種類のフレーズを構成するそれぞれ異なるワードが1つずつ含まれる。上記のとおり、各フレーズおよび異なる各ワードに対応するパラメータフレームのシーケンスは、フレーズアレー115およびワードアレー117の各対応する要素毎に記憶される。

【0056】図13には、2つの「get an image（画像を得る）」のフレーズの発声音を示す音声信号151および153と、「get the earth（接地する）」のフレーズの発声音を示す音声信号155とが示されている。また、図13には、「get（得る）」、「an（1つの）」および「image（画像）」という各ワードの分離的発声音を示す発声信号157、159および161も示されている。更に、図13には、「get an image（画像を得る）」の2つのフレーズの発声音に相当するパラメータフレーム152および154と、「get the earth（接地する）」のフレーズの発声音に相当するパラメータフレーム156と、「得る（get）」、「1つの（an）」および「画像（image）」という各分離ワードの発声音に相当するパラメータフレーム158、160、162も示す。なお、「the（その）」および「earth（接地）」というワードについては、それらが入力された2つ以上のフレーズに含まれていないためワードモデルは生成されない。従って、これら、「その（the）」および「接地（earth）」というワードは、図13には示されていない。

【0057】図14は、各未知ワードのワードモデルを生成するために必要な処理を詳細に示すフローチャートである。ステップS21で、未知ワードを含むフレーズに対応するパラメータフレームシーケンスが、分離して発声された当該未知ワードに対応するパラメータフレームシーケンスと関連付けられる。この関連付け処理には、フレキシブルダイナミックプログラミングアライメント処理が用いられる。このアライメント処理は、そのプロセスの初期段階における未知ワードの始点と終点に関する情報の欠乏を緩和する、特に、各ワードの最適アライメントパスが始まらなければならない場所或いは終わらなければならない場所を制限しないフレキシブルダイナミックプログラミングアライメントプロセスが用いられる。このフレキシブルダイナミックアライメントプロセスについては、ダイナミックプログラミングアライメントについて論じた後に、詳細に後述する。

【0058】このフレキシブルダイナミックプログラミ

ングアライメントの結果、未知ワードに対するパラメータフレームシーケンス内の当該未知ワードのおよその始点および終点が特定される。例えば、「get an image（画像を得る）」というフレーズの第1の発声音に相当するパラメータフレームシーケンス152が、未知ワード「get（得る）」の発声音に相当するパラメータフレームシーケンス158に関連付けられるとき、パラメータフレームシーケンス158内の始点および終点は上記アライメントの結果から確認される。上記のとおり、始点前および終点後のパラメータフレームは、バックグラウンドのノイズまたはフレーズ例におけるワードの発音されない部分に相当しているため、除去される。

【0059】ステップS21において、学習フレーズ例に対して行われるアライメントでは、「get（得る）」というワードの3組の始点および終点と（「get」というワードは3つのフレーズに含まれているため）、「an（1つの）」と「image（画像）」のワードの2組の始点および終点と（「an」と「image」のワードは、2つのフレーズに示されているので）を確認する。ステップS23で、それぞれの未知ワードの平均始点および平均終点が決定され、平均スタートフレームの前にあるフレームおよび平均エンドフレームの後にあるフレームが捨てられる。例えば、「get」というワードをステップS21で処理した後、3つのフレーズ151、153、155を用いて確認された始点がそれぞれフレームf8W1、フレームf9W1、フレームf13W1である場合、その平均はフレームf10W1（ $[8+9+13]/3$ ）である。そして、フレームf10W1の前にあるパラメータフレーム158のシーケンス内のすべてのフレームは、捨てられる。終点についても、捨てられるフレームがエンドフレームを越えるものであるという点を除いて、上記した始点の場合と同様の手順が用いられる。その結果、各ワードのパラメータフレームシーケンスは、上記のワードの分離ワードモデルとなる。

【0060】図15は、「get」、「an」、「image」の各ワードの分離ワードモデル音声信号157'、159'、161'、およびそれぞれに対応するパラメータフレームシーケンス158'、160'、162'を示す。処理のこの段階で、処理された各ワードのステータスフラッグ141（図12）は、FLEXIからCHOPPEDに変わる。なお、CHOPPEDは、未知ワードが始めと終わりが除去されたフレームシーケンスを有していることを示す。

【0061】次に、図14に示されるステップS25において、未知ワードを含むフレーズに対応するパラメータフレームシーケンスは、当該ワードの分離ワードモデルのパラメータフレームシーケンスに関連付けられる。例えば、「get」というワードが出現するフレーズの発声音に対応するパラメータフレームシーケンス152、154、156の各シーケンスと、「get」というワードの分離ワードモデル158'に対応するパラメータフ



フレームシーケンスとの整合がチェックされる。図16は、本例を行った対応付け（アラインメント）の結果を示し、ここで破線は、フレーム間のアラインメントを示す。例えば、パラメータフレームシーケンス152と分離ワードモデルのパラメータフレームシーケンス158'とのパラメータフレームのシーケンスとの間のアラインメントでは、フレームf2P1およびf3P1がフレームf10W1に関連付けられ、フレームf4P1およびf5P1がフレームf11W1に関連付けられる等、アラインメントが確立されている。

【0062】次に、図14のステップS27において、分離ワードモデルの個々のフレームを、フレーズに対応するパラメータフレームのシーケンスからの関連フレームの平均で置き換えることによって未知ワードの標準モデルが生成される。例えば、図16に示される分離ワードモデル158'のパラメータフレームシーケンスにおいて、フレームf10W1は、フレームf2P1、f3P1、f1P2、f2P3の平均に置き換えられる。また、フレームf11W1は、フレームf4P1、f5P1、f2P2、f3P2、f3P3の平均に置き換えられる。分離ワードモデルのフレームの1つに関連付けられるフレームがフレーズに存在しない場合、そのフレームは、隣接する置き換えられたフレームの間を捕間するかまたはフレームから外挿することによって得られるフレームに置き換えられる。例えば、図16に示されるパラメータフレーム158'のシーケンスにおいて、フレームf12W1は、パラメータフレームシーケンス152、154または156内のいずれのフレームにも関連付けられない。従って、f12W1は、f11W1、f13W1の置換後のフレームで捕間することによって得られたフレームに置き換えられる。或いは、分離ワードモデル内における、入力フレーズのいずれのフレームにも関連しないフレームを捨てるように構成しても良い。

【0063】ステップS27で生成された基準ワードモデルは、システムの音声認識部で用いられるものであり、その詳細を以下に示す。なお、処理された各ワードのステータスフラグ141は、CHOPPEDからIN-USEに変化する。

【0064】いったん標準ワードモデルがフレーズ内のすべてのワードについて生成されると、フレーズは、図2に示される言語モデル21に追加される。本実施形態において、言語モデル21は、バイグラムモデル（Bigram Model）と同様で、網目上に相互に接続するノードを備える。ここで、相互接続は、システムに既知となったワードを示す。しかしながら、この相互接続には、例えば、正しい英語の用法に関する文法規則は含まれていない。すなわち、システムに認知されたフレーズに従って、どのワードが他のどのワードに続くかを制限するだけである。図17aは、以下の各フレーズがシステムによって学習されたときに取り出される言語モデル21を

示す。

【0065】

【表1】

get an image	- phrase 1
get the earth	- phrase 2
get the fjord	- phrase 3
get the map	- phrase 4
get the coin	- phrase 5
save an image	- phrase 6
load an image	- phrase 7
make it smaller	- phrase 8
make it larger	- phrase 9
make it brighter	- phrase 10
make it more red	- phrase 11
make it more yellow	- phrase 12
make it more green	- phrase 13
make it more cyan	- phrase 14
make it more blue	- phrase 15
make it more magenta	- phrase 16
quit	- phrase 17

【0066】図17aに示されるように、スタートノードN0、エンドノードNn、8個の中間ノードN1からN8がある。認識されるべき入力フレーズのために、システムは、スタートノードN0からエンドノードNnまでのパスを見つけなければならない。しかしながら、システムは一度学習されると妥当な程度にフレキシブルになり、ユーザが「make it smaller」というフレーズの代わりに「make smaller」と入力しても、システムはその入力されたフレーズを認識することができる。しかしながら、フレーズの個々のワードが既知であっても、そのフレーズがシステムに認知されていない場合、システムはフレーズを認識しない。例えば、上記の言語モデルにおいて、ユーザが「save the image」と言った場合、たとえばシステムにとって「save」、「the」、「image」の各ワードが既知であっても、システムはこの入力フレーズ（「save the image」）を認識しない。

【0067】言語モデル21は、図11のステップS3で入力されたテキストによって示される必要なワードのシーケンスを抽出することによって作成され、各新しいフレーズが入力された後に更新され、フレーズ内の各ワードにはワードモデルが提供される。言語モデル21を更新する方法について、図17bを参照して以下に説明する。

【0068】新しいフレーズが入力され、入力フレーズの各ワードに対応するワードモデルが決定されると、ステップS30においてシステムは、フレーズの第1のワ

ードがスタートノードN0の出力に接続されているかどうかを判断する。接続されている場合、処理はステップS32に進む。他方、第1のワードがスタートノードN0の出力にまだ接続されていない場合は、ステップS31でスタートノードN0からの新しい出力として第1のワードが追加される。

【0069】処理はステップS32に進み、ここでシステムはワードループカウンタwを初期化する。以降の処理で、システムは、このワードループカウンタwを用いて、当該フレーズ内の全ワードを数える。処理はステップS33に進み、ここでシステムはワードwが当該フレーズの最後のワードか否かを判断する。ワードwが最後のワードでない場合、処理はステップS34に進み、ワードwがノード（エンドノードNnを除いた）の入力に接続されているかどうかを判断する。接続されていれば処理はステップS36に進み、ここでシステムは、次のワードw+1がワードwを入力として有するノードの出力に接続されているかどうかをチェックする。他方、ステップS34で、ワードwがいずれのノードの入力にも接続されていないとシステムが判断した場合、処理はステップS35に進み、ここで新しいノードが作成され、ワードwは、新しいノードの入力に接続される。それから、上記のとおり、処理はステップS36に進む。

【0070】ステップS36で、次のワードw+1がワードwを入力として有するノードの出力に接続されていないとシステムが判断した場合、処理はステップS37に進む。ステップS37では、次のワードw+1が当該ノードの出力として追加される。一方、ステップS36で、次のワードw+1が既にワードwを入力として有するノードの出力に接続されているとシステムが判断した場合は、そのままステップS38に進み、ここでワードカウンタwがインクリメントされる。そして、処理はステップS33に戻り、入力フレーズの次のワードに対して同様の処理が行われる。また、ステップS33で、ワードwがフレーズの最後のワードであることを判定された場合、処理はステップS39に進む。ステップS39において、システムは、最後のワードがエンドノードNnの入力に接続されているかどうかを判断する。最後のワードがエンドノードNnの入力に接続されていない場合、処理はステップS40に進み、ここでシステムは、フレーズの最後のワードをエンドノードNnの入力に接続する。最後のワードがエンドノードNnに既に接続されているか、またはステップS40で最後のワードがエンドノードNnに接続されると、本処理を完了し、処理された入力フレーズは言語モデル21の一部を形成することになる。

【0071】本学習プロセスの特徴の1つは、システムに個々にフレーズを学習させることもできるし、一度に多数のフレーズを学習させることもできる。新しいフレーズが入力され、この時点でシステムが既にフレーズ内

のいくつかのワードのワードモデルを有する場合、システムは言語モデル21を更新する前に、当該フレーズの未知ワードについてワードモデルを生成する必要がある。

【0072】＜ダイナミックプログラミング（DP）＞図14のステップS21およびS25で行われる処理において、フレーズのパラメータフレームとワードのパラメータフレームとを関連付けるために、アラインメント処理が用いられた。より効果的な方法で2つのシーケンスを関連させるため、アラインメントプロセスは、例えば、ワードが分離して発せられたり、ワードが連続的に発せられるフレーズに含まれる場合に生じるワードの発声速度の差を補償することが要求される。上記のダイナミックプログラミング（DP）のアラインメントプロセスは、あるワードを別のワードに合致させる一つの方法である。これは、最適な非線形の時間スケールのひずみを適用して、すべての時点で最適にマッチさせる方法で行われる。

【0073】DPマッチングプロセスの概略を、図18から図20を参照して以下に説明する。図18は、横座標方向に入力ワードを示すパラメータフレームのシーケンスを示し、縦座標方向にワードモデルを示すパラメータフレームのシーケンスを示す。本例を図14のステップS25で行われる処理と比較すると、入力ワードを示すパラメータフレームは、入力フレーズの1つに相当するパラメータフレームシーケンスの一部分を示す。また、ワードモデルを示すパラメータフレームシーケンスは、切り出されたワードの1つに相当するフレームシーケンスを示す。

【0074】ワードモデルと入力ワードとのトータル的な違いを見つけるため、図18の左下隅と右上隅との間のあらゆるパスに沿って最短累積距離が得られる数対のフレームに関してすべての距離の和を求める必要がある。この限定は、類似するワードの対応するフレームが正しく関連付けられることを保証する。このような総距離を計算する1つの方法は、すべてのありうるパスを考慮し、各パスに沿った各点の $d(k, j)$ の値（フレームkとフレームjとの間の距離）を加算することである。2つのワードの間で評価された距離の最小値を取り、累積距離を求める。この方法は正しい答えを与えるが、有効パスの数が増加するので、実用的な音声認識システムのためのコンピュータ処理は不可能となる。

【0075】ダイナミックプログラミングは数学的技術であり、すべての取り得るパスの距離を計算することは行わずに、最適なパスの累積距離を求めることができる。累積距離が計算されるべきパスの数は、さらにDPプロセスを制限することによって、低減される。例えば、最適パスは負の傾斜にはなり得ない。負方向の傾斜に進む場合、ワードの1つは他のワードの時間を逆にさかのぼる状態になる。DPプロセスに対するもう1つの

制限は、基準ワードに対する入力ワードの時間圧縮／拡張の最大量を制限することである。本実施形態において、この制限は、マッチングプロセスで飛ばされたりまたは繰り返されるフレームの数を制限することによって、実現される。例えば、図19において、フレームシーケンスは、フレームfkがフレームfjmに合致する場合、フレームfk+1がフレームfjm、fj+1m、fj+2mまたはfj+3mに合致するように制限される。したがって、入力ワードのフレームfkおよびワードモデルのフレームfjmが最適パス上にある場合、上記制限において、最適パス上で直ちに処理する点は、図20に示すように、(k-1, j)、(k-1, j-1)、(k-1, j-2)、(k-1, j-3)のいずれかである。

【0076】図18は、フレームfk-1まで伝わる「有効パス」を示し、入力ワードとワードモデルとの間のマッチング候補を示す。フレームfkが認識ブロック17に入力されるとき、各有効パスは、現フレームfkと有効パスの最後にあるワードモデルのフレームとの間の局部距離を有する。この局部距離は累積距離に追加されるべきものである。多数のパスが同じ点で出会う場合、最

$$D(k, j) = d(k, j) + \min[D(k-1, j), D(k-1, j-1),$$

$$D(k-1, j-2), D(k-1, j-3)] \quad \dots(6)$$

【0082】となる。

【0083】上記制限で、D(0, 0)の値は、d(0, 0)、d(1, 0)、d(2, 0)またはd(3, 0)と等しくなければならず、すべてのパスはこれらの点のうちの1つから始まらなければならない。したがって、始点の1つから始め、D(k, j)の値は、帰納的な処理ルーチンを介して決定され得る。ルーチンがマッチングされるべきワードの終わりに到達すると、DPプロセスによって計算された最小累積距離は、2つのワードをマッチさせる最適なパスのスコアを示す。認識されるべき入力発声音がワードシーケンスから構成される場合、採用された方向を示すバックポインタが用いられなければならない。DPプロセスが最適なパスの終わりを示した後、そのバックポインタで戻りながら跡をたどることによって、入力発声音を認識することができるようになるためである。

【0084】上記のDPプロセスは、すべてのパスを網羅して探索することと比較して、大幅にコンピュータ処理の付加を低減するが、各入力ワードとのマッチングを行うために多数のワードモデルと比較しなければならない場合、他のコンピュータ処理も重要となり得る。したがって、認識結果の精度に大した影響を及ぼさないコンピュータ処理を省略することは望ましい。可能なコンピュータ処理の省略は、スコアの悪いパスが必要以上に形成されることを防ぐことである。成長するパスが木の枝

$$\min[D(k-1, j), D(k-1, j-1), D(k-1, j-2), D(k-1, j-3)] \quad \dots(7)$$

【0088】ダイナミックプログラミングプロセスを実施する方法を、図21～図31を参照して以下に説明す

短の累積距離の有効パスが継続され、他のパスは捨てられる。例えば、図18で、パスA、B、Cは点(k, j)で出会うが、最短累積距離を有するパス(A、BまたはCのいずれか)は継続され、その他の2つのパスは捨てられる。

【0077】したがって、D(k, j)を、ワードの始めから点(k, j)までの有効パスに沿った累積距離とすれば、

【0078】

【数5】

$$D(k, j) = \sum_{\substack{x, y=1 \\ \text{along a} \\ \text{valid path}}}^{k, j} d(x, y) \quad \dots(5)$$

【0079】で表される。

【0080】また、上述した制限は以下のとおりである。すなわち、

【0081】

【数6】

に似ているので、この処理は「刈り込み(pruning)」として知られている。このようにパスの刈り込みによって、最適パスの両側の候補パスを含むパスの狭帯のみが考慮される。このような借り込みが用いられる部分では、ダイナミックプログラミングプロセスが最適なパスを確実に見つけるとは限らないことが認められる。しかしながら、刈り込みしきい値を用いて、例えば5～10の要因でコンピュータ処理の平均量を低減し、ワードがよく類似する場合であれば殆ど常時正しいパスが求められる。

【0085】なお、本実施形態において、図2に示される認識ブロック17は、上記と同様のダイナミックプログラミングマッチングプロセスを用い、ワードモデル19およびノイズモデル23で認識されるべき発声音のパラメータフレームのシーケンスのマッチングを行う。

【0086】＜認識検索＞本実施形態による音声認識システムのもう1つの特徴は、ダイナミックプログラミングプロセスが行われる方法にある。特に、本実施形態は、上記等式(6)で行われた最小値の計算、すなわち、以下の式(7)が処理される現フレームfkに影響されないことを利用する。したがって、前フレームfk-1が処理されるとき、等式(6)のこの部分が計算される。

【0087】

【数7】

る。

【0089】図21は、認識すべき入力発声音を入力し

た際に、認識ブロック17で行われる処理を示すフローチャートである。システムは、プリプロセッサ15によって生成された入力発声音のパラメータフレームシーケンスを処理する。フレームカウンタ変数 $k$ はこの処理に供され、変数 $k$ は、ステップS41で0に初期化され、各フレームが処理される毎にステップS61で順次インクリメントされる。処理される各フレームは、ステップS47で用いられ、各ワードモデル内の他の有効パスの累積距離を更新する。ワードカウンタ $w$ はこの処理に供され、ステップS43で0に初期化され、ステップS47の後、ステップS49でインクリメントされる。ステップS45で、システムは、現フレームを用いて、すべてのワードモデルが処理されたかどうかをチェックする。すなわち、システムは、ワードカウンタ $w$ がシステムに認識されたワードの数 $n_w$ を下回るかどうかをチェックする。

【0090】現フレーム $f_k$ を用いて、全ワードモデルがいったん処理されると、処理はステップS51に進み、ここで図17aに示される言語モデル21のノードが、現フレームを用いて処理される。ステップS51で行われる処理は、現パラメータフレームが、入力音声の始めと終わりまたは入力音声の許されたワードシーケンス間の沈黙に対応するという状況を処理する。また、この処理は、有効パスが、許容されたワードシーケンスの範囲に沿って成長するのを保証する。

【0091】ステップS51でノードが処理された後、各ワードモデルの始め、即ち「エントリステート」の1つで終わる有効パスの累積距離がステップS57で更新される。この処理は、現パラメータフレーム $f_k$ があるワードモデルの終わりにマッチングするときに、次のパラメータフレーム $f_{k+1}$ が他のワードモデルの始めに合致するという状況に対処する。これを達成するため、ワードカウンタ $w$ は、ステップS53で再度初期化される。ステップS55でシステムは、すべてのワードモデルが処理されているかどうかをチェックする。ステップS57でシステムは、現ワードモデルのエントリステートに対する累積距離を更新し、ワードカウンタ $w$ が、ステップS59でインクリメントされる。そして処理は、ステップS55に戻る。

【0092】すべてのワードモデルが現パラメータフレーム $f_k$ について処理された後、パラメータフレームカウンタ変数 $k$ は、ステップS61でインクリメントされる。ステップS63でシステムは、処理されるべき入力発声音がこれ以上存在するかどうかを判断する。これはステップS63で変数 $k$ をシステム限界(LIMIT)及び音声終了識別子(EOS)と比較することによって、この処理は行われる。システム限界は、図3に示されるADC63から出力された音声サンプルを記憶するために用いられるバッファの容量によって決定される。

【0093】入力発声音のすべてのパラメータフレーム

が処理された場合、DPプロセスが完了し、回帰アルゴリズムを用いて最適パスを決定し、それによって、認識結果を決定する。他方、ステップS63でシステムが処理されるべきパラメータフレームがさらに存在すると判断された場合、ステップS65でシステムは刈り込みしきい値を調整し、ステップS43に戻る。刈り込みしきい値 $Th$ は、次の入力フレームが処理される時、ステップS47、S51、S57で処理される有効パスの数を制限するべくステップS65で調整される。

【0094】図21のステップS47で行われる処理の詳細を、ワードモデルの特定の例を示す図22から図26を参照して以下に説明する。図22は、ワードモデル201の一例を示す。特に例示されているワードモデル201は、学習セッションの間に得られたステート $S_0 \sim S_9$ のシーケンスと、ワードモデル201の終わりの出口ステート $SD$ とを備える。これらの目的を以下に説明する。

【0095】ワードモデル(WORD)201の各ステート $S$ は、そのステートで終わる有効パスの累積距離を記憶する累積距離記憶 $D[S]$ を有する。本実施形態において、ワードモデル201は、現フレーム $f_k$ の現アクティブリスト(ACTIVE LIST)203を有する。ここで、現アクティブリストは、現フレーム $f_k$ の有効パスの終わりにあるワードモデルのステートを降順に登録する。したがって、現アクティブリスト203の各ステートは、そのステートで終わる各有効パスの累積距離を記憶する。この例において、現フレーム $f_k$ の現アクティブリスト203は、ステート $S_7$ 、 $S_5$ 、 $S_4$ 、 $S_3$ 、 $S_2$ 、 $S_1$ 、 $S_0$ を登録している。現アクティブリスト203に掲載されたステートはアクティブステートとして示される。本実施形態において、ワードモデル201は、関連付けられた新アクティブリスト(NEW ACTIVE LIST)205も有する。この新アクティブリスト205は、ステップS47で行われる処理の間に完成し、次のフレーム $f_{k+1}$ のための有効パスの終わりにおけるワードモデル201のステートを登録する。

【0096】現アクティブリスト203および新アクティブリスト205の重要性を図23を参照して説明する。図23は、 $p_1 \sim p_7$ の7つの有効パスを示し、7つの有効パスは、入力されたワードとワードモデル201との間の現フレーム $f_k$ までの7つのマッチング候補を示す。図示されるとおり、7つの有効パス $p_1 \sim p_7$ はそれぞれ、ワードモデル201のステート $S_7$ 、 $S_5$ 、 $S_4$ 、 $S_3$ 、 $S_2$ 、 $S_1$ 、 $S_0$ で終わっている。現アクティブリスト203で降順に登録されているのは、これら有効パスの終わりのステートである。新アクティブリスト205に登録すべきステートを決定するため、すなわち、次の入力フレーム $f_{k+1}$ に残るパスを決定するため、ある入力パラメータフレームから他のパラメータフレームへの移行が許可されている状態を考慮しなければ

ならない。

【0097】入力発声音に対する標準モデルの時間圧縮の最大量は、入力発声音の隣接するフレーム間においてスキップできるステートの最大数によって決定される。本実施形態において、この最大数は2に設定される。すなわち、DPプロセスは、図19に示される状態遷移図に従う。入力発声音に対する標準モデルの時間伸長の最大量は、同じステートにマッチし得る連続した入力フレームの最大数によって決定される。本実施形態においては、続いて入力される2つの連続フレームが同じステートにマッチし得る。これらの状態を監視するため、各ステートSは、関連する自己反復カウンタ、すなわち、SELFを有する。SELFは、対応する有効パスが入力フレームfkから次のフレームfk+1が同じステートで終わる度にインクリメントされる。したがって、例えば、パスp5は、図23に破線で示されるパス207の1つまたはすべてに沿って伝わる。図23に示される他のパスp1～p4およびp6、p7も、同様の方法で伝わる。2つ以上のパスが同じ点で合う場合、最小の累積距離を有するパスが維持され、他のパスは捨てられる。さらに、パスの累積距離が刈り込みしきい値より大きい場合、このパスも捨てられる。このように、新しいパスが連続的に作成され、一方、他のパスは捨てられる。刈り込みしきい値の目的は、各入力パラメータフレームのために処理される有効パスの数を制限し、それによって、アルゴリズムに必要な時間量およびメモリを制限することである。

【0098】図24は、図21のステップS47で行われる処理ステップを詳細に示す。まず、ステップS71でポインタLAが初期化される。また、ワードモデル201の出口ステートに記憶される累積距離、すなわちD[SD]は、大きい値を示すHUGEに設定される。ポインタLAは、新アクティブリスト205に登録された最後のアクティブステートを示すのに用いられる。最初、新アクティブリスト205にはアクティブステートがなく、従って、ポインタLAは、出口ステートSDを示すように設定される。ステップS73でシステムは、現アクティブリスト203にアクティブなステートがあるかどうかをチェックする。言い換えれば、現ワードにおける、現フレームfkで終わる有効パスがあるかどうかチェックする。本例では、現アクティブリスト203には7つのアクティブステートがあり、システムは各アクティブステートを順次処理する。カウンタ変数iが与えられ、カウンタ変数iは、現アクティブリスト203におけるアクティブステートすべてをカウントするために用いられる。カウンタ変数iは、ステップS75で0に設定され、現アクティブリスト203の全アクティブステートが処理されるまで、ステップS79でインクリメントされる。システムは、ステップS81でカウンタ変数iと現アクティブリスト203のアクティブステートの数naとを比較することによって、すべてのアクティブ

ステートが処理されたかどうかを判断する。

【0099】現アクティブリスト203のすべてのアクティブステートがいったん処理されると、ステップS77の処理の間に発生された新アクティブリスト205が、ステップS83にて、処理されるべき入力発声音の次のフレームfk+1の現アクティブリスト203に置き換えられる。実際には、2つのアクティブリストを示すために用いられるポインタを交換することによってこの処理は行われる。古くなった現アクティブリストは、次の入力フレームfk+1の処理中に新アクティブリストとして上書きされる。最後に、ステップS85において、アクティブステートになり、新アクティブリスト205に登録された、ポインタLAによって示される最終ステートが図21に示されるステップS57で用いるために格納される。

【0100】ステップS77で行われる処理の概略を、アクティブステートS7、S5を例に挙げて説明する。図23に示されるように、アクティブステートS7、S5はそれぞれ、パスp1、p2の終わりである。図25は、2つの有効パスp1、p2の一部分を示し、有効パスp1、p2はそれぞれ、現フレームfkのアクティブステートS7、S5で終わる。図25の破線は、各2つのパスp1、p2が次のフレームfk+1に伝わり得る態様を示す。破線213で示すように、パスp1は、フレームfk+1で他のワードに拡張することができる。したがって、(アクティブステートS7に記憶される)パスp1の累積距離は、出口ステートSDに複写される。破線215、217、219で示すように、パスp1は、ステートS9、S8、S7のそれぞれにも伝わり得る。したがって、パスp1の累積距離がステートS9、S8、S7に複写される。更に、ステートS9、S8、S7は、降順に新アクティブリスト205に追加され(ただし、入力フレームと比較されず、当該ワードを離れる全てのパスの最短累積距離を記憶するためだけに用いられる出口ステートは追加されない)、最終アクティブポインタLAは、最後に追加されたステート(すなわち、ステートS7)を示すように設定される。

【0101】図26aは、現アクティブリスト203の第1のステートS7が処理された後の新アクティブリスト205を示す。図示されるように、ステートS9、S8、S7はそれぞれ、新アクティブリスト205の最初の3つの要素で、最終アクティブポインタLAは、ステートS7を示す。

【0102】図25をもう一度参照すると、破線221、223、225、227で示すように、パスp2は、ステートS8、S7、S6、S5それぞれに伝わり得る。しかしながら、2つのステートS8、S7は次のフレームfk+1のために記憶された累積距離をすでに有するので、パスp2の累積距離を単に各ステートにコピーするという処理だけではない。すなわち、これら2つのス

テートS8、S7では、すでに記憶されている累積距離とパス $p_2$ に関連する累積距離とを比較し、最小の累積距離がこれら2つのステートにコピーされることになる。言い換えば、図23に示されるパスのステートS8、S7に記憶されるべき累積距離は、アクティブステートS5を処理した後に、 $\min(D[S7], D[S5])$ によって求められる。一方アクティブステートS5に記憶された累積距離はステートS6に直接に複写される。ステートS6には、次のフレーム $f_{k+1}$ に対する累積距離が記憶されていないからである。2つのステートS6、S5は、新アクティブリスト205に追加され、最終アクティブポインタLAは、ステートS5を示すように設定される。

【0103】図26bは、現アクティブリスト203の第2番目のアクティブステートS5が処理された後の新アクティブリスト205を示す。図示されるように、ステートS8、S7、S6、S5はそれぞれ、新アクティブリスト205の最初の5つの要素となり、最終アクティブポインタLAは、ステートS5を示す。現アクティブリスト203の他のアクティブステートも同様の方法で処理され、処理は図21で示されるステップS49に進み、ここでワードカウントがインクリメントされる。

【0104】最終アクティブポインタLAが設けられているので、システムは比較処理を必要とするステートおよび比較を必要としないステートを確認するために新アクティブリスト205を検査する必要がある。即ち、ステートがアクティブポインタLAによって示されたステート番号と等しいかまたは上回る場合、比較処理が必要であることがわかる。また、ステート番号よりも下回る場合、累積距離がそのステートにコピーされればよい。

【0105】図24に示されるステップS77で行われる処理を、図22に示されるワードモデル201の例を用いて、図27aおよび図27bを参照して詳細に説明する。ステップS77で $i=0$ の場合に、処理をしなければならない第1のアクティブステートSは、ステートS7である。したがって、図27aのステップS91で、システムは、ステートS7で終わる有効パスの累積距離が刈り込みしきい値 $Th$ を下回るかどうかをチェックする、すなわち、 $D[S7]$ は $Th$ と比較される。 $D[S7]$ が刈り込みしきい値 $Th$ を上回る場合、このパスは捨てられ、処理は、図24に示されるステップS79に移行する。 $D[S7]$ が刈り込みしきい値 $Th$ を下回る場合、処理は、ステップS92に進み、ここで現フレーム $f_k$ のために処理されるアクティブステートの総数を数えるために用いられる変数ACOUNTをインクリメントする。ステップS93で、システムは、処理中である現アクティブステートS7と処理中である現フレーム $f_k$ との間の局部距離を計算し、この計算結果を累積距離 $D[S7]$ に加算する。

【0106】本実施形態においては、以下のユークリッ

ドの距離公式を用い、現フレーム $f_k$ と現アクティブステートSとの間の局部距離の値を得る。

【0107】

【数8】

$$d(s, f_k) = \|S - f_k\|^2 \quad \dots(8)$$

【0108】累積距離 $D[S7]$ がステップS93で更新された後、ステップS95でシステムは、現アクティブステートS7で終わる有効パスが次の入力フレーム $f_{k+1}$ でワードから離れ得るかどうかをチェックする。上記DPの制限によって、この処理では、処理される現アクティブステートSの先の3つ目のステートがワードモデル201の最後のステートの先に進むかどうかを決定して示す。本例では、現アクティブステートS7の先の3つのステートが最後のステートS9を過ぎているので、ステップS97に進み、ワードモデル201の終わりである出口ステートSDを示すようにポインタjが設定される。ステップS101で、ポインタjによって示されるステートが最終アクティブポインタLAによって示されるステートと比較される。この処理は（現時点では）、処理されるべき現アクティブリスト203内の第1番目のアクティブステートであるので、最終アクティブポインタLAは、出口ステートSDを示している（図24に示されるステップS71を参照）。したがって、処理はステップS103に進み、ここでポインタjによって示されたステート、すなわち、出口ステートSDに記憶された累積距離が、処理される現アクティブステートS7で終わる有効パス $p_1$ の累積距離と比較される。

【0109】図24のステップS71で、出口ステート、すなわち、 $D[SD]$ に記憶された累積距離は、大きな値を示すHUGEに設定されている。したがって、累積距離は、処理される現アクティブステートS7に記憶されている累積距離を上回る。結果的に、処理は、ステップS105に進み、ここで $D[S7]$ が $D[SD]$ にコピーされる。次に、ステップS107で、ポインタjによって示されるステート、すなわち、出口ステートSDに対応する自己反復カウンタは0に設定される。ステップS109で、ポインタjはデクリメントされ、ステートS9を示すようになり、処理はステップS101に戻る。

【0110】次に、ポインタjによって示されるステート（ステートS9）は、最終アクティブポインタLAによって示されるステート（ステートSD）の前なので、処理は図27bに示されるステップS111に進む。ステップS111で、システムは、ポインタjによって示されるステート、すなわち、ステートS9が処理される現アクティブステートS7と等しいかどうかをチェックする。ここでは、等しくないので、処理はステップS113に進み、ここでステートS9に記憶された累積距離が現アクティブステートS7に記憶された累積距離と等

しくされる。言い換えれば、パス $p1$ の累積距離がステート $S9$ に複写される。そして、ステップ $S115$ で、ステート $S9$ が、図22に示される新アクティブリスト205に追加される。ステップ $S117$ で、ステート $S9$ に対応する自己反復カウンタが0に再設定される。そして、続くステップ $S119$ でポインタ $j$ がデクリメントされ、ポインタ $j$ はステート $S8$ を示す。処理はステップ $S111$ に戻り、ステート $S8$ はステート $S9$ と同様の方法で処理される。

【0111】ステート $S8$ がステップ $S113$ 、 $S115$ 、 $S117$ で処理された後、 $j$ はステップ $S119$ でデクリメントされてステート $S7$ を示すようになる。ステート $S7$ は処理される現アクティブステートでもある。したがって、このとき、処理はステップ $S111$ からステップ $S121$ に進み、ここでステート $S7$ に関連する自己反復カウンタがチェックされる。すなわち、ステート $S7$ で終わる有効パスが、当該入力発声音の過去の2つのフレームに渡ってそこで終わったかどうかをチェックする。ステート $S7$ に対応する自己反復カウンタの値が2つのフレームと等しい場合、ステート $S7$ は新アクティブリスト205には追加されず、処理はステップ $S123$ に進む。ステップ $S123$ では、最終アクティブポインタ $LA$ が、現アクティブステートに1を加えられた値に設定され、ステート $S8$ を示すようになる。このルーチンによって、現アクティブステートに対応する自己反復カウンタが2つのフレームと等しいかどうか確認され、等しい場合には現アクティブステートで終わる有効パスが、次のフレーム $f_{k+1}$ の同じステートに伝わることを防ぐ。図23に示されるパスの例においては、現フレーム $f_k$ のステート $S7$ で終わるパス $p1$ はステート $S6$ から入るので、自己反復カウンタは0となっている。結果的に、ステート $S7$ は、ステップ $S125$ で新アクティブリスト205に追加される。ステート $S7$ の自己反復カウンタは、ステップ $S127$ でインクリメントされ、ステップ $S129$ で、最終アクティブポインタ $LA$ が現アクティブステート、すなわち、ステート $S7$ を示すように設定される。

【0112】ステップ $S129$ またはステップ $S123$ の後、処理はステップ $S131$ に進み、ここで現アクティブステート $S7$ に記憶された累積距離は、現フレーム $f_k$ のために処理されたすべてのワードにおけるすべての有効パスの最短累積距離 $MINSCORE$ と比較される。ステート $S7$ に記憶された累積距離が $MINSCORE$ を下回る場合は、ステップ $S133$ で $MINSCORE$ がステート $S7$ に記憶された累積距離に置き換えられる。その後、処理は、図24に示されるステップ $S79$ に戻る。一方、 $MINSCORE$ に記憶された累積距離が現ステート $S7$ に対応する累積距離を下回る場合、処理は、そのまま図24に示されるステップ $S79$ に戻る。ステップ $S79$ に戻ると、カウンタ変数 $i$ がインクリメントされ、現アクティブリスト

203の次のアクティブステート、すなわち、ステート $S5$ が、ステップ $S77$ で処理されることになる。

【0113】アクティブステート $S5$ は、上記アクティブステート $S7$ と同様の方法で処理される。ステート $S5$ で終わる有効パスの累積距離が刈り込みしきい値 $Th$ を下回る場合、ステップ $S93$ で、システムは、現アクティブステート $S5$ と処理される現フレーム $f_k$ との間の局部距離を計算し、その計算結果を現アクティブステート $S5$ に記憶された累積距離に加算する。ステップ $S95$ で、システムは、現アクティブステート $S5$ で終わるパス $p2$ が、上記 $DP$ の制限によって、次のフレーム $f_{k+1}$ の他のワードに拡張できないと判断する。言い換えれば、ステート $S5$ から3つ目のステートは、ステート $SD$ と等しくないかまたはステート $SD$ を越えないので、処理はステップ $S99$ に進み、ここでポインタ $j$ が、ステート $S5$ の3段階先のステート、すなわち、ステート $S8$ を示すように設定される。次に、ステップ $S101$ で、ポインタ $j$ によって示されるステートが、最終アクティブポインタ $LA$ によって示されるステートと比較される。

【0114】この時点において、最終アクティブポインタ $LA$ はステート $S7$ を示し、ポインタ $j$ はステート $S8$ を示す。したがって、処理がステップ $S103$ に進み、ここで（アクティブステート $S7$ の処理の結果として）ステート $S8$ にすでに記憶されていた累積距離が、アクティブステート $S5$ に記憶された累積距離と比較される。ステート $S8$ に記憶された累積距離がステート $S5$ に記憶された累積距離を上回る場合、ステート $S8$ に記憶された累積距離は、ステート $S5$ に記憶された累積距離に置き換えられる。ステップ $S107$ で、ステート $S8$ に関連する自己反復カウンタは0に再設定され、ポインタ $j$ は、ステップ $S109$ でデクリメントされ、ステート $S7$ を示すようになる。そして、処理はステップ $S101$ に進み、ここで上記と同様の処理が行われる。

【0115】この再帰処理ルーチンは、システムに認知されたすべての標準ワードの全ての現アクティブステートについて行われる。

【0116】現フレーム $f_k$ について上記方法で各ワードを処理した後、言語モデル21の各ノードが順次処理される。上記のとおり、言語モデル21は、許容されるワードのシーケンスを決定する。この情報は、ノードによって、特にノードの入力および出力に接続されるワードによって、決定される。図21のステップ $S51$ のノードの処理によって、有効パスは確実に許容されたワードシーケンスの中を伝わる。以下、ステップ $S51$ で行われる処理を、図28を参照して詳細に説明する。

【0117】まず、ノードを処理する前に、バックグラウンドのノイズを示すフレームと現フレーム $f_k$ （すなわち、 $d(\text{noise}, f_k)$ ）との間の局部距離が、ステップ $S151$ で計算される。ステップ $S153$ で、ノードボ



インタvがスタートノードN0を示すように初期化される。ステップS155で、ノードポインタvによって示されるノードに記憶された累積距離、すなわち、D[v]が、刈り込みしきい値Thと比較される。D[v]が刈り込みしきい値Thを下回る場合、処理はステップS157に進み、ここでd(noise, fk)が、処理される現ノードvに記憶された累積距離に加算される。ステップS159で、システムは、D[v]と最小値記憶MINSCOREに記憶された値を比較する。D[v]がMINSCOREに記憶された値を下回る場合、ステップS161でD[v]をMINSCOREにコピーする。カウント値ACOUNT(現フレームのために処理されたアクティブステートおよびノードの数を示す)はステップS163でインクリメントされ、処理はステップS165に進む。一方、ステップS155において、D[v]が刈り込みしきい値Thを上回る場合は、ステップS167でD[v]は大きな値を示すHUGEに設定され、処理はステップS165に進む。

【0118】ステップS165、S168で行われる処理について、図29に示されるノードNを例にして説明する。図示のノードNには、3つのワード「get」、「save」、「load」がその入力に接続され、2つのワード「an」、「the」がその出力に接続されている。このようなノードの生成は、図17bに示される手順を用いては実現できないが、ダイナミックプログラミングプロセスがより複雑な言語モデルのために動作することを示すために本例が選択されている。特に、定型文法では、ノードが図29に示されるようなものが一般的である。

【0119】ステップS165で、システムは、ノードNの入力に接続されたワードの出口ステート、すなわち、ワード「get」、「save」、「load」の出口ステートに記憶されたすべての累積距離の最小値を決定する。一般的な場合、この計算は以下の式によって、示される。

【0120】

【数9】

$$\min_{(v_w[v])} (D[S_D]) \quad \dots (9)$$

【0121】ここで、Iw[v]は、ノードvの入力に接続されたすべてのワードを示す。システムがノードNについてこの最小累積距離を決定した後、その最小累積距離がすでに記憶されている累積距離を下回る場合には、当該最小累積距離がノードNに記憶された累積距離D[N]に複写される。事実上、これは、当該ノードの入力に接続されたワードの一つであって、現在そのノードにおいて成長しているパスの累積距離よりも小さい累積距離を有する有効パスが存在するかどうかの判断である。

【0122】バックグラウンドのノイズフレームにマッチするギャップが、フレーズ内の各ワードの手前、間及

び終わりに存在し得るので、有効パスはそのノード内を伝わるができる。図29の矢印231が示すように、有効パスが、入力されたフレームから次の入力フレームまでの間1つのノードに残ることもできる。この場合、有効パスはノードNを出発してノードNに戻る。ワードモデルのステートと異なり、パスは、連続して入力されたフレームのいくつ分でも、ノード内に残ることが可能である。システムがステップS165の処理を行った後、ノードNに記憶された累積距離がすでに記憶されている値より小さい場合、ノードNに記憶された累積距離は、ステップS168で、一時的記憶INSCOREに複写される。この一時的記憶INSCOREは、ワード「an」、「the」のためのボックス233、235で示される。2つ以上のワードがノードの出力に接続されるので、比較処理が行われなければならない。接続ワードに伝わるのは最小累積距離を有するパスだけだからである。図21に示されるステップS57の処理において、ワードの一時的記憶INSCOREに記憶された累積距離を用いて当該ワードのエントリステートが更新される。

【0123】ステップS169で、システムは、D[v]がHUGEに等しいかどうかをチェックする。等しい場合、終了する有効パスが無いかまたは、ノードvを通過して次のフレームfk+1に接続されたワードに入っている有効パスが無いことを示す。D[v]が値HUGEを下回る場合、有効パスが、ノードvで終わっているかまたは、ノードvを通過して、次のフレームfk+1でノードvに接続されているワードに入っていることを示す。したがって、次のフレームfk+1における潜在的にアクティブなステート(およびノード)の数を表すカウンタPA COUNTは、そのノードに関連する沈黙ステートが次のフレームfk+1でアクティブステートとなり得るので、ステップS171でインクリメントされる。ノードポインタvは、ステップS173でインクリメントされ、言語モデル21の次のノードを示すようになる。システムは、ステップS175で、ノードポインタvが言語モデル21のエンドノードNnを越えるノードを示すかどうかをチェックすることによって、言語モデル21のすべてのノードが処理されているかどうかをチェックする。システムがすべてのノード処理を完了していない場合、処理はステップS155に戻る。一方、処理を完了している場合、処理は図21に示されるステップS53に戻る。

【0124】次に、図21に示されるステップS57で行われる処理を、図22に示されるワードモデル201、図30および図31を参照して、詳細に説明する。図30において、ステップS181でシステムは、INSCOREに記憶された累積距離がHUGEと等しいかどうかをチェックする。等しい場合、次の時点でこのワードに入る有効パスがないことを示す。したがって、このワードを再処理する必要がないので、処理はステップS207に

進み、ここで次の入力フレーム  $f_{k+1}$  のアクティブステートの数（図24のステップS83の処理によって現アクティブリスト203に現在記憶されているステートの数となる）が、カウントPACOUNTに加算される。その後、処理は、図21に示されるステップS59に戻り、ここでワードカウントがインクリメントされ、次のワードモデルが処理される。

【0125】他方、INSCOREがHUGEと等しくない場合は、有効パスが先のワードを出て、処理すべき現ワードに入っていることを示す。したがって、別のワードモデル（エントリステート）から延びてきたパスによって到達され得る現ワードモデルのステートは、INSCOREに記憶された累積距離を用いて更新されなければならない。上記DP制限を用いる本実施形態において、エントリステートは、ステートS0、S1、S2である。この更新手順は、以下の方法で行われる。まず、ステップS183で、システムは、処理される現ワードを示すワードモデルが（出口ステートは含まない）3つより多いステートを含むことをチェックする。3つより多いステートがある場合、ステップS185でステートS2を示すようにステータスポインタjが設定される。他方、現ワードのステートが3つより少ない場合、ステップS187でステータスポインタjが、当該処理ワードの終わりの出口ステートSDを示すべく設定される。このように、ステップS187またはS185でステートSD、S2のいずれかに向かうようにステータスポインタjが設定されると、処理はステップS189に進み、ポインタjによって示されたステートと最終アクティブポインタLAによって示されるステートとが比較される。

【0126】図27aおよび27bにおいてステップのシーケンスで行われた処理のように、ポインタjによって示されるステートが、アクティブポインタLAによって示されるステートを越える場合、そのステートに既に記憶された累積距離とINSCOREに記憶された累積距離が比較される。

【0127】図23に示されるDPパスの例として、パスp7は、次のフレーム  $f_{k+1}$  でステートS1、S2、S3に伝わり得るが、ステートS0には伝わらない。パスp7が先の2つのフレームにおいてステートS0で終わっているからである。したがって、最終アクティブポインタLAは、ステートS1を示すようになる（ステップS121、123）。

【0128】図31は、図22に示されるワードモデル201のエントリステート（すなわち、最初の3つのステート）を示す。図示されるように、アクティブポインタLAは、ステートS1に向かう。ワードモデル201に3つよりも多くのステートがあるので、ステータスポインタjは、ステートS2を示す。ステップS189で、システムは、ポインタjによって示されたステートが、最終アクティブポインタLAによって示されたステ

ート、すなわちステートS1を過ぎているかどうかを判断する。したがって、ここでは、処理はステップS191に進む。ステップS191で、システムは、ステートS2に記憶された累積距離と、ワードモデル201に関連する一時的記憶INSCOREに記憶された累積距離とを比較する。ワードモデル201の一時的記憶INSCOREは、図31に示される矩形のボックス241で示される。INSCOREに記憶された累積距離がステートS2に記憶された累積距離より小さい場合、ステップS193でINSCOREに記憶された累積距離がステートS2に複写される。そして、ステップS195で、ステートS2の自己反復カウンタが0に再設定され、処理はステップS197に進む。一方、INSCOREに記憶された累積距離がステートS2に記憶された累積距離より大きい場合、ステートS2に記憶された累積距離は変化せず、処理はステップS197に進む。ステップS197において、ポインタjはデクリメントされ、ステートS1を示すようになる。処理はステップS189に戻り、同様の処理が、ステートS1について行われる。

【0129】ステートS1について処理した後、ポインタjは、ステップS197で再びデクリメントされ、ステートS0を示すようになる。したがって、処理はステップS189からステップS198に進み、ここでシステムは、処理すべきステートがあるかどうかをチェックする。本例の場合、ステートS0がまだ処理されるべきであるので、処理はステップS199に進む。ステップS199において、INSCOREに記憶された累積距離が、ステートS0に複写される。このステートは、最終アクティブポインタLAによって示される最後のアクティブステートの手前なので、ステートS0について累積距離の比較は行われない。ステップS201で、システムは、ステートS0を現アクティブリスト（このリストは、図24のステップS83の処理の前では新アクティブリスト205だったものである）に追加する。そして、ステップS203で、ポインタjはデクリメントされ、ステートS1を示すようになる。その後、処理はステップS198に戻り、ここでシステムは、処理されるべき現ワードにエントリステートがこれ以上ないと判断する。従って処理はステップS205に進み、ここで対応する一時的記憶INSCOREに記憶された累積距離が、HUGEに再設定される。そして、ステップS207で、現アクティブリストのステートの数が、カウントPACOUNTに追加され、処理は、図21に示されるステップS59に戻る。

【0130】＜刈り込み（Pruning）＞図21によれば、ステップS63で、処理されるべき入力フレームが更に存在するとシステムが判断した場合、処理はステップS65に進み、ここで刈り込みしきい値Thが調整される。刈り込みを用いる目的は、ある時点からつぎの時点まで伝わるDPパスの数を制限することである。特

に、本実施形態では、刈り込みしきい値を調整し、実際に処理されるアクティブステートの数を前もって決められた範囲に制限できるようにすることを狙う。なお、この限られた範囲は、利用できる動作メモリの量および処理時間によって左右される。さらに、本実施形態の目的は、高価なコンピュータ処理のオーバーヘッドを必要とせず、以上の目的を達成させることである。本実施形態において、刈り込みしきい値は、可変差分値 (PRUNING) を処理された入力フレームのために決定されたすべての最小累積スコア MINSCORE に追加することによって、決定される。すなわち、刈り込みしきい値は、等式 (10) によって、求められる。

【0131】

【数10】

$$Th = MINSCORE + PRUNING \quad \cdots (10)$$

【0132】アクティブステートの設定数のみが各入力フレームのために処理されることを確実にする方法の1つは、そこに記憶された累積距離を増加させるために、処理されつつある入力フレームに対するすべてのアクティブリストにあるアクティブステートをソートし、最小累積距離を有する入力フレームで始まる所望の数を処理する。しかしながら、この技術は、アクティブステートをソートするために長時間のコンピュータ処理時間を要する。コストのかかるコンピュータ処理によるソートを行うのではなく、本実施形態で用いられる技術は、最後の入力フレームを処理した後に利用できる情報を用いる。特に、本実施形態において、差分値 (PRUNING) は、処理されるべき次の入力フレームの潜在的なアクティブステートの数 (PACOUNT に格納されている) に依存して変化し、実際に処理されるステートの数を2つのしきい値の間にあるように維持する。刈り込みしきい値  $Th$  が調整される方法を、図32を参照して詳細に説明する。

【0133】ステップS211において、システムは、処理される次のフレームの潜在的アクティブステートの数 (PACOUNT に記憶される) とステートしきい値 (STATE TH) を比較する。ここで、ステートしきい値 (STATE TH) は、利用できる動作メモリの量によって決定される絶対的最大のステートしきい値を下回るが、その値の付近に設定される。PACOUNT に記憶された値が STATE TH を下回る場合、すべての潜在的なアクティブステートが処理され得ることを意味するので、直前の時点で用いられた差分値 PRUNING は増加される。したがって、ステップS213で、調整数  $dp1$  が既存の差分値 PRUNING に追加される。 $dp1$  の値は、妥当な局部距離を上回るように設定されるので、すべてではないがほとんどの潜在的アクティブステートが処理される。

【0134】PRUNING に記憶された値は、ステップS215で、高い刈り込みしきい値 HIGHPRTH と比較される。上限が差分値 PRUNING に設定されているので、これ以上

進む必要のない最大差分値が存在する。PRUNING に記憶された値が HIGHPRTH を下回る場合、処理はステップS219に進む。一方、PRUNING に記憶された値が HIGHPRTH を上回る場合、PRUNING は、ステップS217で、HIGHPRTH に設定される。ステップS215またはS217の処理の後、システムは、差分値 PRUNING を追加された他の有効パスの最小累積距離 MINSCORE と PRUNING との和に等しくなるように刈り込みしきい値  $Th$  が設定される。そして、処理は、図21に示されるステップS43に戻る。

【0135】ステップS211で、次のフレームの潜在的アクティブステートの数 (PACOUNT) が STATE TH を上回るとシステムが判断した場合、ステップS221で、システムは、直前の入力フレームの処理においてアクティブステートとして処理されたステートの数 (ACOUNT に記憶されている) と、低ステートしきい値 LOWSTH とを比較する。この LOWSTH の値は、ACOUNT が LOWSTH を下回った場合には、多くの時間またはメモリ量を費やさずに次の入力フレームのすべての潜在的なアクティブステートを処理できることを保証できるような値に設定される。したがって、ACOUNT が LOWSTH を下回る場合、処理はステップS221からステップS213に進み、上述したような処理によって差分値 PRUNING が調整される。他方、ACOUNT が LOWSTH を上回る場合、すべての潜在的アクティブステートが処理された場合に、それほど処理時間またはメモリ量を費やさずに済むという保証は無い。したがって、差分値 PRUNING を低減する必要があるかもしれない。

【0136】差分値 PRUNING を低減させる必要があるかどうかを判断するため、システムは、ステップS223で、ACOUNT と STATE TH を比較する。ACOUNT が STATE TH を下回る場合、システムは、差分値 PRUNING が HIGHPRTH と等しいかどうかをチェックする。差分値 PRUNING が HIGHPRTH と等しい場合、システムはすべてのアクティブステートを処理するように試みてきたことを示す。したがって、次の入力フレームのために処理されるアクティブステートの数によって、長時間の処理時間または大容量メモリを要する処理になってしまうようなことはまず起こらないとみてよい。したがって、差分値 PRUNING を変化させずにステップS219に処理を進め、ここで刈り込みしきい値は、MINSCORE に差分値 PRUNING を加算したものと等しくなるように設定される。他方、差分値 PRUNING が HIGHPRTH を下回らない場合 (この場合、差分値 PRUNING が HIGHPRTH を下回ることになる)、次の入力フレームのために処理されるアクティブステートの数は、長時間の処理時間または大容量のメモリを費やす可能性がある。したがって、処理されるアクティブステートの実際の数が計算されなければならない。この計算は、変更させていない差分値 PRUNING を用いてステップS231で設定された刈り込みしきい値を用いてステップS233

で行われる。

【0137】ステップS223に戻って、ACOUNTがSTATETHを上回るとシステムが判断した場合、ステップS225で、差分値PRUNINGは、調整定数 $d p 1$ によって低減される。ステップS225で差分値PRUNINGが減らされた後、システムはステップS227において、差分値PRUNINGが低刈り込みしきい値LOWPRTHを下回るかどうかを判断する。低刈り込みしきい値は、次の入力フレームについて処理されるアクティブステートの数が設定された非常しきい値EMGSTHを上回ることを保証するのに用いる。この処理を行う理由は、大量に刈り込まれた場合、ダイナミックプログラミングプロセスが行えないからである。差分値PRUNINGが低刈り込みしきい値LOWPRTHを下回る場合、ステップS229で差分値PRUNINGをLOWPRTHに等しくする。ステップS231では、調整された差分値PRUNINGをMINSOREに加算したもの等しくなるように刈り込みしきい値 $T h$ が設定される。次にステップS233で、システムは、次の入力フレームのために処理されるアクティブステートの数をカウントする。この処理は、すべてのアクティブステートに記憶された累積距離と、新たに決定された刈り込みしきい値 $T h$ を有するすべてのノードに記憶された累積距離とを比較することによって行われる。

【0138】総数( $n s a$ )は、次の入力フレームに関して処理されるアクティブステートとノードの総数を示す。総数 $n s a$ が非常しきい値EMGSTHを下回る場合、刈り込みしきい値の設定が低すぎることを意味する。従って、処理はステップS213に進み、ここで差分値PRUNINGが増加され、刈り込みしきい値 $T h$ が再設定される。一方、 $n s a$ が非常しきい値EMGSTHを下回らない場合、ステップS237で、 $n s a$ はLOWSTHと比較される。 $n s a$ がLOWSTHを上回る場合、ステップS231で設定された刈り込みしきい値 $T h$ は受け入れ可能であり、処理は、図21に示されるステップS43に戻る。他方、 $n s a$ がLOWSTHを下回る場合、刈り込みしきい値は増加させることができる。よって、ステップS219で刈り込みしきい値 $T h$ が再設定される前に、ステップS239で、第2の調整定数 $d p 2$ を差分値PRUNINGに加算する。本実施形態において、この第2の調整定数 $d p 2$ は、調整定数 $d p 1$ の半分に設定される。

【0139】当業者には明らかなように、刈り込みしきい値を変化させる方法はコンピュータ処理に高価なコストを費やさせない。さらに、各時点で処理されるアクティブステートの数を制限するという方法を用いて刈り込みしきい値が調整される。このため、割り当てられる処理時間およびメモリが限界を越えないように調整することができる。

【0140】図21に示される処理ステップのシーケンスを用いて入力シーケンスのすべてのフレームが処理された後、ダイナミックプログラミングプロセスによって

決定された最適パスにより正しいパスを決定するためにバックトラッキングルーチンが要求される。本実施形態において、バックトラッキングルーチンは、各パスが伝わるワードのシーケンスを示すバックポイントを追跡する。このようなバックトラッキングルーチン(再帰ルーチン)が行われる方法、およびポイントを発生させる方法の詳細については、音声処理の当業者によく知られているので、これ以上説明はしない。

【0141】<初期化>システムに入力発声音を認識させる前に、認識処理中に用いられるシステムのしきい値および変数を初期化しなければならない。この処理は、以下の方法で行われる。まず、スタートノードN0に記憶される累積距離は0に設定され、他のノードに記憶される累積距離は、大きな値を示すHUGEに等しくなるように設定される。各ワードモデルに関連付けられ、潜在的なアクティブステートの数をカウントするカウンタPACOUNTは0に設定される。各ワードモデルに関連付けられた最終アクティブポイントLAは、それぞれのモデルのエンドステートSDを示すように設定される。各ワードモデルに関連付けられた一時的記憶INSOREは、大きい値を示すHUGEに設定される。その後、すべてのノードが処理され、各ワードの入力に接続されるすべてのノードの累積距離の最小値は、各ワードに関連付けられた一時的記憶INSOREに複写される。これは、スタートノードN0に接続された各ワードの一時的記憶INSOREが0に設定されることを保証する。最後に、各ワードのINSOREに記憶された値を用いて、各ワードモデルのエントリステートを作動させ、初期化する。各ワードモデルのエントリステートを初期化する処理ステップは、図30を参照して上述したエントリステートを更新するために用いられる処理ステップと同一である。刈り込みしきい値および差分値PRUNINGもまた、第1の入力フレームを処理する前に、初期化される。特に、刈り込みしきい値 $T h$ は大きい値を示すHUGEに設定され、差分値PRUNINGは、高刈り込みしきい値HIGHPRTHと等しくなるように設定される。

【0142】<フレキシブルダイナミックプログラミングアラインメント>上記において、本実施形態におけるワードモデルの生成方法について、図13から図16を参照して詳細に説明を行った。特に、先ずはじめに、フレーズに含まれるワードについて分離ワードモデルが、フレキシブルダイナミックプログラミングアラインメントを用いて生成される。すなわち、フレキシブルダイナミックプログラミングアラインメントを用いて、そのフレーズに対応するパラメータフレームのシーケンスを、分離して発声された当該フレーズに含まれるワードに対応するパラメータフレームに関連付けることで分離ワードモデルが生成される。以下、このフレキシブルアラインメントプロセスについて詳細に説明する。なお、ここでは、学習フレーズ「get an image」が与えられ、か

つ、そのフレーズに存在するワードに対するワードモデルが無い場合を例に揚げて説明する。

【0143】図33は、「get an image」というフレーズの発声音声に対応するパラメータフレームシーケンス152と、「get」、「an」及び「image」の各分離ワードの発声音声に対応するパラメータフレームシーケンス158、160及び162を示す。パラメータフレームシーケンス152におけるパラメータフレームの幾つかはバックグラウンドノイズもしくは沈黙に対応するので、フレーズに含まれる各分離ワードの間、先頭の分離ワードの開始部分、最終の分離ワードの後部分にノード251、253、255、257が提供される。これらのノードは図17aで示された言語モデル中のノードに類似の態様で動作し、処理中であるパラメータフレーム152の現パラメータフレームが沈黙もしくはバックグラウンドノイズに対応したものであるという状況に対応可能である。この可能性は、図33において、ノード251、253、255及び257における沈黙フレームf sil（これは図10において示されたノイズモデル23である）によって示されている。

【0144】パラメータフレームシーケンス158、160、162のそれぞれの開始部分もしくは終了部分におけるフレームのいくつかは沈黙或いはバックグラウンドノイズに対応するものであるかもしれないが、パラメータフレームシーケンス152内の沈黙或いはバックグラウンドノイズに対応するパラメータフレームは、シーケンス158、160及び162内の沈黙に対応するフレームよりも、ノード251、252、255及び257に格納された沈黙フレームf silによりよくマッチする。これは、沈黙フレームf silが全沈黙フレームの平均を示すものであり、シーケンス152内の沈黙に対応するフレームと沈黙フレームf silとの間の変動量が、シーケンス152内の沈黙に対応するフレームと各シーケンス158、160及び162内の沈黙に対応するフレームとの間の変動量よりも小さいからである。

【0145】フレキシブルアラインメントプロセスが実行される方法は、図18～32を参照して説明したような、入力音声格納されている標準モデルに関連付けられる方法に類似している。特に、パラメータフレーム158、160及び162を標準モデルとして、ノード251、253、255、257、そしてシーケンスフレーム152を入力されたフレームとして用いる場合、フレキシブルアラインメントプロセスの全体的な処理ステップは図21に示されたフローチャートに従う。混乱を避けるために、分離的に発声されたワードを表すパラメータフレームシーケンス158、160、162は、ステートと称する。未知の入力音声を認識するために用いられた標準モデルのステートと同様に、これらのステートも関連する累積距離格納部を有する。この累積距離格納部は、処理中のシーケンス152の現フレームに関

して、当該ステートまでのダイナミックプログラミングパスの累積距離を格納する。

【0146】フレキシブルアラインメントプロセスと未知の入力発声音声を認識するために用いられたアラインメントプロセスとの主要な相違は次の通りである。すなわち、フレキシブルアラインメントにおいては、(i)各ダイナミックプログラミングパスがあらゆる位置で（エントリステートの一つに限らず）ワードに入ることができる、(ii)各ダイナミックプログラミングパスはそこに存在するあらゆるステートから出ることができる。

【0147】フレキシブルアラインメントプロセスの動作を上記例を用いて以下に説明する。なお、以下の説明は、シーケンスパラメータフレーム152のはじめの少しのパラメータフレームについて行う。

【0148】まず、第1のフレームを処理する前に、ノード及びワードモデル内のステートの累積距離スコアを初期化する。この初期化処理手順は、上述の、未知入力音声を認識するのに先立って実行される初期化処理に類似している。特に、スタートノード、すなわちノード251に格納される累積距離はゼロに設定され、他の全てのノードに格納される累積距離は大きな値を示すHUGEに設定される。その後、ワードW1、W2及びW3におけるステートの累積距離スコアは、それらのワードの入力に接続されたノードに格納された累積距離スコアを用いて更新される。これは、フレームf0P1が処理されているときに、ダイナミックプログラミングパスが第1のワードW1及び第1のノード251の各ステートからスタートできることを保証する。

【0149】初期化の後、第1フレームf0P1が、ワードW1、W2及びW3に関して順次処理される。しかしながら、ワードW2及びW3内のステートに関連した累積距離は値HUGEを有するので、第1ワードW1内のステートに関してのみ、第1フレームが処理されることになる。ワードW1に関して第1フレームを処理する際、フレームf0P1とワードW1内の各ステートとの間の距離は各ステートに対応付けられた累積距離格納部に格納される。フレキシブルアラインメントプロセスは図28に示された処理ステップを用いて、ノード251、253、255及び257を順次処理する。最後に、ノード処理の結果を用いてワードW1、W2及びW3内のステートの累積距離スコアを更新することにより、第1フレームf0P1の処理が完了する。この更新処理は、エントリステート（すなわち、最初の3つのステート）だけでなくワード内の全てのステートが更新されるということを除けば、図30に示される処理と同様である。

【0150】シーケンス152の第1のパラメータフレームが処理されると、第1のパラメータフレームf0P1の処理によってスタートされたダイナミックプログラミングパスを成長させるために第2のパラメータフレーム

f1P1が処理される。入力された発声音の認識に用いられたダイナミックプログラミング手法で行ったように、各ワードW1、W2及びW3のステートは逆方向の順序で処理される。ここで、本実施形態において、図19及び図20を参照して記述された伝搬の制限に類似した制限が用いられる。相違点は、未知の入力音声の認識する場合では最後の3つのステートからのみ現ワードを抜けることが許されたが、あらゆるステートから現ワードを抜けることができるという点のみである。ワードW1、W2及びW3が第2のパラメータフレームf1P1を用いて処理されると、各ノード内で現在伝わっているダイナミックプログラミングパスを更新するために、ノード251、253、255及び257が処理される。この処理がなされると、ノード処理の結果を考慮するため、ダイナミックプログラミングパスを更新するために、ワードW1、W2及びW3が再び処理される。

【0151】パラメータフレームシーケンス152の残りのパラメータフレームも、同様の方法で順次処理される。シーケンス152内の全てのパラメータフレームが処理されると、最も低い累積スコアを有するダイナミックプログラミングパスが決定される。パラメータフレームシーケンス158、160及び162内の開始及び終了フレームを特定するために、フレキシブルアライメントプロセスの間の各ダイナミックプログラミングパスの成長が記録される。ここで、これら開始及び終了フレームは、対応するワードを表す（沈黙は表さない）それらパラメータフレームの境界を示す。

【0152】特に、ダイナミックプログラミングパスが手前のワードから、あるいはワードの手前からワードへ入ったときはいつも、そのダイナミックプログラミングパスが入ったステートが記録され、そのパスに関連付けられる。同様に、ダイナミックプログラミングパスがワードから出て、それが出たステートが記録されたとき、そのパスに関連するスコアはそのワードから出た全ての前のダイナミックプログラミングパスに関連するスコアよりも低い。従って、パラメータフレームシーケンス152内の最後のパラメータフレームが処理され、最良のスコアを有するダイナミックプログラミングパスが特定されると、パラメータフレームシーケンス158、160、162内のワードのおおよその開始及び終了位置が、ダイナミックプログラミングパスに関連する記録を参照することにより、特定される。

【0153】当業者には明らかなように、フレキシブルアライメントプロセスに関する上記記載は、入力フレーズ中に含まれる各ワードについてワードモデルが存在しないという状況に特化されたものである。しかしながら、標準モデルの学習は自然に増加するように、すなわちユーザが適宜システムに学習させることができるように設計されているので、入力フレーズ中のワードに対するワードモデルが既に存在するという状況が発生し得

る。この場合、フレーズとこのワードとの間のアライメントの最中において、既にワードモデルを有するワードに対しては通常のダイナミックプログラミングプロセスを適用し、まだワードモデルの存在しない他のワードに対してはフレキシブルダイナミックプログラミングアライメントを適用するという、混合型のダイナミックプログラミングアライメントプロセスが採用される。

【0154】図14のステップS25、S26を参照して説明したように、フレーズ内の未知のワードに対して分離ワードモデルが決定されると、それらはワードを含む入力フレーズのパラメータフレームシーケンスに関連付けられ、ワードモデルがその結果から生成される。

【0155】＜ワードモデルの適用＞本実施形態による音声認識システムの他の特徴は、図10のワードモデル19、ノイズモデル23および言語モデル21が構築／更新モジュール91によって更新され、修正されることである。したがって、記憶されたワードモデル19は、別のユーザの入力音声によって、修正されたり適応させられたりする。

【0156】図34は、ワードモデル19が別のユーザに適用される方法を示すフローチャートである。ステップS251で、新しいユーザは、既知のワードまたはフレーズをマイクロフォン7またはキーボード3を介してシステムに入力する。したがって、構築／更新モジュール91は、新しいユーザから入力された発声音に対応するパラメータフレームのシーケンスと、キーボード3を介して入力された対応するテキストとを有することになる。ステップS253で、システムは、入力発声音と、当該発声音に存在することが認知されているワードの既存のワードモデルとを、ダイナミックプログラミングルーチンを用いて関連付ける。すなわち、ダイナミックプログラミングルーチンは入力発声音のパラメータフレームと適切なワードモデルとを関連付ける。ステップS255で、システムは、ワードモデルのステートをそのワードモデルに関連付けられたパラメータフレームのシーケンスで直接に置き換える。ステップS257で、新しいユーザが他のフレーズを入力することを決定した場合、処理はステップS251に戻り、同様のルーチンが次の入力発声音のために再度行われる。ステップS257で、新しいユーザがこれ以上フレーズを用いないと決定した場合、当該処理は完了する。

【0157】図35は、ワードモデル19が別のユーザのために用いられる第2の方法を示すフローチャートである。ステップS261で、新しいユーザは、マイクロフォンを介して複数回およびキーボードを介して1回、既知のワードまたはフレーズをシステムに入力する。したがって、構築／更新モジュール91は、パラメータフレームの複数のシーケンスおよびキーボード3を介して入力された対応テキストを有するようになる。ここで、これら複数のシーケンスは、新しいユーザによって発声



された既知のワードまたはフレーズに対応する。ステップS263で、システムは、各入力発声音と、当該発声音に含まれることが既知であるワードに対応する既存のワードモデルとを、ダイナミックプログラミングルーチンを用いて関連付ける。

【0158】ダイナミックプログラミングルーチンは、各入力発声音のパラメータフレームと、適正なワードモデルのステートとを関連付ける。ステップS265で、システムは、ワードモデルのステートをワードモデルに関連付けられたパラメータフレームの平均で置き換える。本実施形態において、ワードモデルのステートが発声音のパラメータフレームのいずれにも関連付けられなかった場合、システムは、置き換えられた隣接ステートを用いて挿入または外挿する。ステップS267で、新しいユーザが他のフレーズを用いることを決定する場合、処理はステップS261に戻り、同様のルーチンが次のフレーズに対して再度行われる。ステップS267で、新しいユーザがこれ以上フレーズを入力しないことを決定すると、処理は完了する。したがって、当業者に明らかなように、新しいユーザは、ユーザの便宜上、既存のワードモデルを増加させながら用いることができる。さらに、新しいユーザは、新しいワードまたはフレーズを上述した方法でシステムに追加することもできる。

【0159】＜他の実施形態＞多数の変形が、本発明の進歩性の概念から逸脱することなく、上記音声認識システムになされ得る。これら種々の変形例のいくつかを以下に説明する。

【0160】上記実施形態においては、すべての発声音が処理される前に受信されるが、本システムは、音声を受信しながら処理するというように、ワードモデルを漸進的に動作させることができる。このような実施形態において、入力バッファが必要であるが、1つのフレームに相当する入力音声、すなわち、20msecの音声を記憶しさえすればよい。当業者には明らかなように、本システムを動作させるため、入力された音声の1つのフレームに対する処理のすべて（プリプロセッサおよび認識ブロックで行なわれる）は、入力音声のその次のフレームについて、処理のための準備が整う前に完了していなければならない。上記フレーム速度およびフレーム継続時間を用いれば、入力音声のフレームを処理するために費やされる時間は10msecを下回るにちがいない。これは現在のプロセッサで達成し得るものである。さらに、入力発声音の各フレームのパワーパラメータは、異なる方法で正規化されなければならない。このような実施形態におけるパワーの正規化の方法の1つとしては、例えば過去の20個にわたる入力音声のパワーに基づいて適応される適応正規化係数を用いることが挙げられる。

【0161】また、第1の実施形態において、ダイナミ

ックプログラミングパスの終わりにあったワードモデルのステートは、そのワードモデルに対応するアクティブリストに登録されていた。他の実施形態においては、総括アクティブリストを1つ備え、そこに、すべてのワードモデルのすべてのアクティブステートを登録する。このような他の実施形態においては、総括アクティブリストに関連して、特定のアクティブステートがどのワードモデルに属するかの情報が記憶されなければならない。

【0162】第1の実施形態において、ワードモデルのステートは、持続時間において、認識されるべき入力音声のフレームと対応する。他の実施形態において、ワードモデルの各ステートは、持続時間に関して、例えば入力音声の3つの連続フレームと等しい。このような他の実施形態において、入力フレームは、3つのフレームグループによって平均化され、ワードモデルのステートに関連付けられる。

【0163】さらに、他の実施形態は、ワードモデルに統計モデル、例えば隠れマルコフモデルを採用するものであり、これは音声認識の当業者に知られている。このような実施形態においては、入力発声音とワードモデルのシーケンスとの間の最小累積距離を決定するのではなく、隠れマルコフモデルの特定のシーケンスによって入力シーケンスが生成された最大の可能性が決定される。このような実施形態において、隠れマルコフモデルは、第1の実施形態で発生された連続標準モデルと同様の方法で発生される。特に、ワードの分離標準モデルは、ワードの発声音とそのワードを含むフレーズの1つ以上の発声音とを比較することによって、生成される。平均パラメータフレームおよび隠れマルコフモデルのステートの共分散マトリクスを生成し、ステートの遷移確率を生成するために、分離標準モデルはワードを含む複数のフレーズ例とともに用いられる。この処理が行われる方法は、音声認識の当業者には明らかである。

【0164】第1の実施形態において用いられる標準モデルは、すべてのワードに対応する。当業者に明らかなように、このことは本質的なことではない。標準モデルは、ワードの部分、例えば、音節、複数のワード、個々の音素にさえ相当する。しかしながら、音素に相当する標準モデルを用いると、システムが言語に依存するようになり、不利である。すべてのワードに対応する標準モデルは、すべてのフレーズに対応する標準モデルより好ましい。なぜなら、時間およびコンピュータ処理の手間を節約することができるからである。特に、フレーズ内のワードをモデルにし、言語モデルを用いることによって、システムに少量のワードしか用いない多くの異なるフレーズを教えられる。他方、標準モデルがすべてのフレーズに相当する場合、標準モデルは、システムによって学習されるべき種々のフレーズのそれぞれに必要となる。また、この利点に加えて、ワードに相当する標準モデルを用いると、フレーズのワード間の間隔に対するシ



システムの柔軟性が増す。これは、フレーズの始めまたは終わりおよびフレーズのワードの間に現れる環境モデルによって可能になる。

【0165】さらに他の実施形態において、モデルの連続フレームが類似する場合、標準モデルは圧縮される。標準モデルが圧縮されると、類似の連続フレームは、1つのフレームに置き換えられる。このような実施形態において、入力発声音の連続フレームが2回までしかワードモデルの同じ状態に関連付けられないというような動的プログラミングプロセスに対する制限は、除去されなければならない。

【0166】図17に示される言語モデルにおいて、あるワードに2つの異なるワードが続く場合、それら2つのワードの何れが当該ワードに続くかについて優先度はない。他の実施形態においては、いくつかのワードのシーケンスに他のシーケンスより優先度を高めるよう、重みを付けることができる。例えば、図17aに示されるフレーズについては、「make it more…」(ここで、「…」の部分には色名称が続く)というフレーズは、「make it smaller」または「make it larger」または「make it brighter」というフレーズより一般的であることが知られているとする。従って、この場合は、ノードN7からノードN8までの遷移が、ノードN7からノードNnまでの遷移よりも強くなるように設定される。これは、ノードN7から「more」、「smaller」、「larger」、「brighter」というワードへ伝わる累積距離に重み付けを行う重み係数を用いることによって達成される。

【0167】当業者には明らかなように、許容されたワードのシーケンスを決めるために用いられる言語モデルは、バイグラム(二重字)モデルである必要はないが、いかなる知られたタイプの言語モデル、例えば定型文法モデルでもよい。用いられた言語モデルのタイプが変更される場合、上記の動的プログラミングマッチングプロセスにはいくつかの変更が必要となるが、このような改造は、音声認識の当業者には明らかである。しかしながら、マッチングプロセスの重要な特徴部分は変化されないままとなる。どんなパターンの合致プロセスにも合うように設計されているからである。

【0168】第1の実施形態において、あるワードの標準モデルを生成する前に、そのワードを含む少なくとも2つのフレーズが入力されなければならない。これは、操作上の好ましい方法であり、ワードモデルを、そのワードを含む1つのフレーズ例のみから各ワード毎に発生させることもできる。しかしながら、このような実施形態によって得られる標準モデルは、どのようなフレーズが与えられた場合でも、あまり典型的なものとはならない。さらに、第1の実施形態で用いられた学習方法において、1つのワードに対していったん標準モデルを決定すると、ワードモデルは、次の入学習フレーズがその

ワードを含むかどうかに関らず、変更されない。他の実施形態においては、すでにワードモデルがあるワード例を含む入力フレーズを用いて、学習セッション中に既存のワードモデルを更新することを可能とする。

【0169】ユーザが音声認識システムを学習させて、既に分離して入力された複数のワードを含むフレーズを入力すると、システムは、それらのワードについて分離した状態で再度入力するようユーザに促すことはしない。これに対して、他の実施形態においては、システムは、ユーザにそのようなワードを再度入力するよう促し、ワードの2つの発声音が大して違わないことを確認するべくチェックを実行する。

【0170】第1の実施形態において、標準モデルが学習されたりまたは用いられる場合、ユーザはテキストを入力し、それに相当する音声コマンド入力しなければならない。音声認識の当業者には分かるように、これは必須ではない。テキストを入力する代わりに、ユーザはキーボード、ファクシミリ、複写機等の相当するキーまたはキーの組み合わせを押すようにしてもよい。例えば、複写機で用いるためにシステムに学習させる場合において、音声コマンド「copy(複写)」と入力し、これに対する所望の機械の応答は、ユーザが複写ボタンを押さえれば、入力される。

【0171】さらに、上記の動的プログラミングマッチングプロセス、標準モデルの発生および適応プロセスを実行させる方法が、他のタイプのパターンマッチングのためにも用いられることは、パターンマッチングの当業者には明らかである。例えば、上記パターン合致プロセスは、手書き文字の認識または他のパターン合致技術で用いられることが考えられる。

【0172】連続ワード音声認識システム(文音声認識システム)は、上記の第1の実施形態に記載されているが、記載されたシステムの多くの特徴が他の種類の音声認識システムにも等しく適用できることは当業者には明らかである。例えば、分離ワードモデルを決定する方法および上記分離ワードモデルは、分離ワード音声認識システム(単語音声認識システム)で用いられる。同様に、動的プログラミングプロセスが行われる方法の多くの特徴および刈り込みしきい値が調整される方法は、いずれの音声認識システムでも用いられ、ここで標準モデルは、音声のいかなる部分、例えば音素、音節等にも対応させ得る。

【0173】第1の実施形態に記載された音声認識システムは、多くの異なるソフトウェアアプリケーション、例えばスプレッドシートパッケージ、グラフィックパッケージ、ワードプロセッサパッケージ等とともに用いることができる。音声認識システムがそのような複数のソフトウェアパッケージとともに用いられる場合、特に、各アプリケーションで用いられるフレーズが異なる場合、各パッケージ毎にワードおよび言語モデルを分ける

ことが有利である。この理由は、ワードモデルの数が増加したり、言語モデルのサイズが大きくなるにつれて、システムが入力発声音を認識するための時間が長くなるからである。したがって、各アプリケーション毎にワードおよび言語モデルを分けることによって、音声認識システムの速度を維持できる。さらに、いくつかのワードおよび言語モデルは、各アプリケーションに用いられ得る。

【0174】さらに、当業者に明らかなように、上記音声認識システムは、多くの異なるタイプのハードウェアでも用いられる。例えば、パーソナルコンピュータ等の機器で用いられるのは明らかであるが、それ以外に、音声認識システムは、ファクシミリ、電話、プリンタ、複写機等、人間と機械のインタフェースを有する各種機器のユーザインタフェースとして用いられ得る。

【0175】＜適用例＞上記音声認識システムの適用の1つは、コンピュータグラフィックパッケージのユーザインタフェースである。グラフィックパッケージのユーザは、複雑なポインティングデバイスまたはキーボードによるグラフィックメニューの操作を用いることなく、マイクロフォンを介して画像操作コマンドを入力できる。

【0176】図36は、音声認識システムのユーザインタフェースを有する画像編集システムのブロック図である。特に、システムは、ユーザからの音響音声コマンドをそれに相当する電気信号に変換するマイクロフォン7と、入力音声を認識するための、例えば上述したような音声認識装置(SPEECH RECOGNITION)271と、認識音声を解釈するコマンドインタプリタ(COMMAND INTERPRETER)273と、コマンドインタプリタに応答するグラフィックパッケージ(GRAPHIC PACKAGE)275と、現在編集されている画像を表示するディスプレイ(DISPLAY)277とから構成される。

【0177】入力音声コマンドは音声認識ブロック271によって認識され、コマンドインタプリタ273によって解釈される。コマンドインタプリタは、ディスプレイ277に表示された画像を編集するための指示シーケンスをグラフィックパッケージ275に送る。グラフィックパッケージ275は、ユーザからの他の経路からの指示やデータを受け取ることができるが、これは矢印279によって示されている。

【0178】本システムは、表示された画像の色操作に役立つ。このようなシステムの機能を、図37および図38を参照して説明する。図37は、海291、空293、雲295、陸地297の風景図を示す。ユーザが海291をもっと青くしたい場合、ユーザは「make this more blue (これをもっと青くする)」というコマンドを入力し、マウスのようなポインティングデバイスを用いて、ディスプレイ277上の海291を指示する。システムは入力を認識し、グラフィックパッケージ275

は表示画像を変化させ、所望の変化を行う。図36に示されるシステムがこの処理を行う方法を、以下により詳細に説明する。

【0179】ユーザがマイクロフォン7を介して「make this more blue」というコマンドを入力すると、音声認識装置271は発声音を表す代表的なパターンシーケンスに変換し、そのシーケンスを予め記憶されている既知の、音声を示すパターンシーケンスと比較する。この比較から、音声認識装置271は認識結果をコマンドインタプリタ273に出力し、コマンドインタプリタ273は認識結果を解釈し、例えばハードディスクに記憶された適当なプログラムを起動する。本例において、起動されたプログラムは、グラフィックパッケージ275に接続されたポインティングデバイス(図示されていない)の監視を開始し、青くしなければならない対象の詳細を受け取る。ユーザは、ポインティングデバイスによって制御されるカーソルを海291が表示されている領域に移動し、ポインティングデバイスの選択ボタンを押し、その押下状態を保つ。選択ボタンの押下状態を保ちながら、ユーザは、カーソルを、ディスプレイ277の海291に対応する領域の全域で移動させる。カーソルが海291全体を移動するにつれて、コマンドインタプリタ273によって起動されたプログラムは、色相の値、カーソルによってたどられる画素の彩度および明度を収集する。ユーザが選択ボタンを放すと、プログラムは、低い値の彩度または明度を持たないサンプル画素の平均色相と標準偏差を決定する。

【0180】図38は、6時の位置が0°及び赤色に相当し、10時の位置が120°及び緑色に相当し、2時の位置が240°及び青色に相当する色相チャート301を示す。上記の例において、プログラムが平均色相を40°、標準偏差を5°に決定したとすると、その結果、色相値の範囲は、図38に示される色相チャートの領域303によって示される。海291をもっと青くするため、システムは、領域303内の色相値を有するすべての画素を図38に示される目標領域305に変化させる。目標領域305は次のようにして決定される。まずサンプル画素のために決定された平均色相値と青に相当する色相値との間の最小弧を決定し(この弧は図38の弧307によって示される)、この弧の角度を4つに分割し、最後に本例のように、領域303内の色相値を有するすべての画素の色相値に分割した結果の角度を加えたり、或いは減じることによって目標領域が決定される。本例において、平均色相は40°であり、したがって、弧の角度が200°なので、領域303内にあるすべての画素の色相値から50°が減じられる。画像は再表示され、海291はユーザに「より青い」を示す。

【0181】当業者には分かるように、与えられた色に相当する特定の色相値を持たせるために、領域303にある色相範囲を有するすべての画素を変化させることが

できる。1つのパレットの色をユーザが作ったパレットの色に関連付けることができるので、ユーザは、機器にユーザが定義したパレットの1つをロードすることを依頼すれば良い。もとの画像の色は、ユーザ定義パレットの相当する色に変化される。代わりに、同様の処理技術を用いて、画像の一部または複数部分の明度または彩度を変更することができる。この処理は、例えば出力がファクシミリ機器からの白黒画像であり、ユーザが再生画像の部分の色調のコントラストを強調したい場合にも好適である。上記の実施形態では画像再生にディスプレイを用いているが、システムがファクシミリ機器または複写機とともに用いられるような場合、このようなディスプレイはオプションであることは明らかである。グラフィック操作システムはコンピュータに関連して説明されているが、カメラ、特にディジタルモーション/スチールカメラのような画像装置にも適用できる。

【0182】以上のように、ユーザは、色編集用語または色操作ツールを用いることなく、画像を編集できる。

【0183】また、本発明は上記実施形態によって限定されるものではなく、種々の変形および実施形態が当業者には明らかである。

【0184】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0185】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0186】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0187】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0188】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0189】さらに、記憶媒体から読出されたプログラ

ムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0190】

【発明の効果】以上説明したように本はつめによれば、色操作を表現する音声入力によって画像操作を行うことが可能となり、容易に所望の画像操作が行える。

【0191】

【図面の簡単な説明】

【図1】本発明の一実施形態を操作するためにプログラムされ得るコンピュータの概略図である。

【図2】音声認識システムの概略構成を示すブロック図である。

【図3】図2に示されたシステムの一部であるプリプロセッサの構成を示すとともに、入力音声信号に行われる処理ステップを示すブロック図である。

【図4】入力音声信号S(t)を一連の時間フレーム(FRAME)に分割した状態を示す図である。

【図5】1つの時間フレームの典型的な音声信号例を示す線図である。

【図6】図5に示される音声信号の離散フーリエ変換の振幅応答を示す図である。

【図7】メルスケールフィルタバンクの平均振幅応答出力を示す図である。

【図8】メルスケールフィルタバンクからの出力の対数振幅スペクトルを示す図である。

【図9】図8に示される対数振幅スペクトルのセプストラムを示す図である。

【図10】学習処理中に用いられる標準モデル構築部の恒星を示すブロック図である。

【図11】音声認識システムで用いるワードモデルを発生させるための学習処理中に行われる処理を示すフローチャートである。

【図12】学習処理中に学習フレーズおよび学習ワードが記憶される一形態例を示す図である。

【図13】複数の入力フレーズ及び入力ワードと、それらのフレーズ及びワードに対応するパラメータフレームシーケンスを示す図である。

【図14】学習処理中に入力された入力発声音からワードモデルを決定するための処理を示すフローチャートである。

【図15】端の部分が切り捨てられた学習ワード及びそれに対応するパラメータフレームシーケンスを示す図である。

【図16】図15に示されるワードのパラメータフレームと、そのワードが含まれる入力フレーズのパラメータ

フレームとの関連付けを示す図である。

【図17A】複数の入力フレーズ例を学習処理している間に発生された言語モデル例を示す図である。

【図17B】システムが新しいフレーズを言語モデルに追加する方法を示すフローチャートである。

【図18】ダイナミック処理技術を用いて入力ワードがワードモデルに関連付けられる際の処理を説明する図である。

【図19】ある入力フレームから次の入力フレームのステートへの許可された移行シーケンスを示す図である。

【図20】図19に示される許可された状態移行シーケンスの他の例を示す図である。

【図21】第1の実施形態で用いられるダイナミックプログラミングアラインメント技術の実行手順を示すフローチャートである。

【図22】ワードモデル、現アクティブリストおよびそれらに関連する新アクティブリストを示す概略図である。

【図23】標準モデル内で伝わるダイナミックプログラミングパスの複数の例を示す図である。

【図24】図21のステップS47に含まれる処理を示すフローチャートである。

【図25】図23で示された2つのダイナミックプログラミングパスが現入力フレームから次の入力フレームに伝わる方法を説明する図である。

【図26A】図22に示されるワードモデルの現アクティブリストの第1の状態が処理された後の新アクティブリストの内容を示す図である。

【図26B】図22に示されるワードモデルの現アクティブリストの第2の状態が処理された後の新アクティ

ブリストの内容を示す図である。

【図27A】図24のステップS77で行われる処理を示すフローチャートである。

【図27B】図24のステップS77で行われる処理を示すフローチャートである。

【図28】図21のステップS51で行われる処理を示すフローチャートである。

【図29】図28に示される処理中において、ノードNに対して行われる処理を示す図である。

【図30】図21のステップS57で行なわれる処理を示すフローチャートである。

【図31】図22に示されるワードモデルのエントリステートを示す図である。

【図32】図21のステップS65で行われる処理を示すフローチャートである。

【図33】入力フレーズのパラメータフレームシーケンスと、入力フレーズ内に含まれるワードの分離的発声によるパラメータフレームシーケンスとを示す図である。

【図34】他の実施形態による第1の代替技術を用いて、ワードモデルを異なるユーザに適用するための処理を示すフローチャートである。

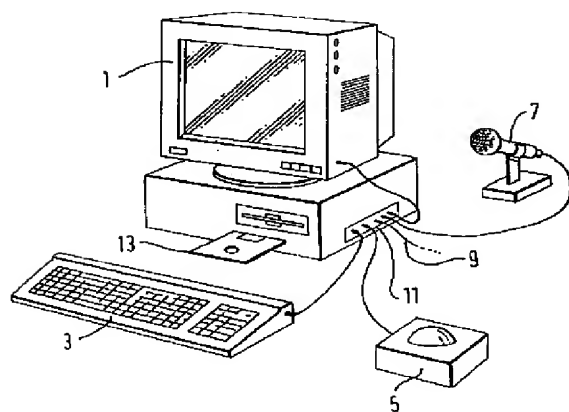
【図35】他の実施形態による第2の代替技術を用いて、ワードモデルを異なるユーザに適用するための処理を示すフローチャートである。

【図36】音声認識ユーザインタフェースを有する画像編集システムのブロック図である。

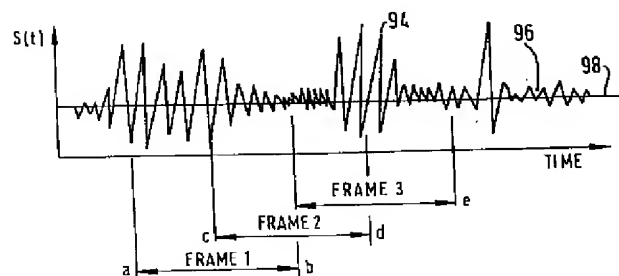
【図37】海、空、雲および陸地を含む風景図の表示例を示す図である。

【図38】円周上に表される色相チャートを説明する図である。

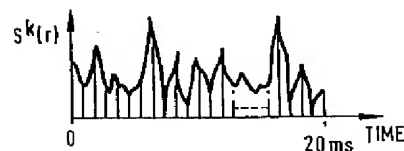
【図1】



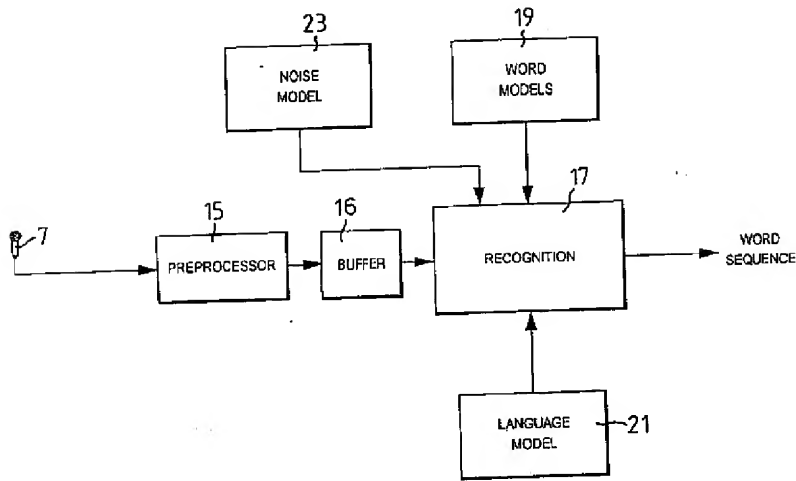
【図4】



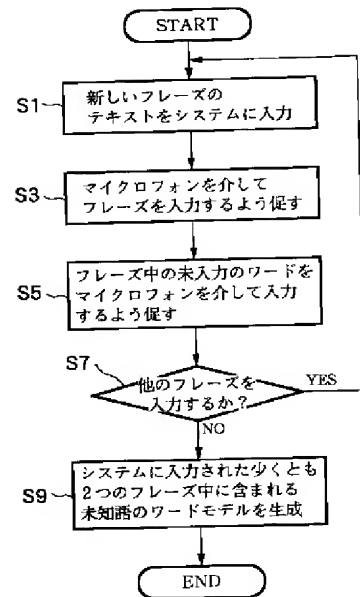
【図5】



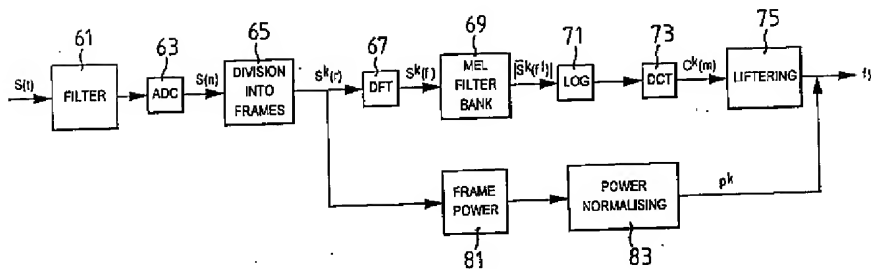
【図2】



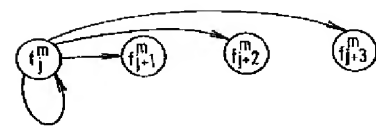
【図11】



【図3】

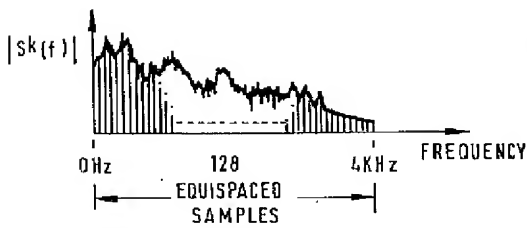


【図19】

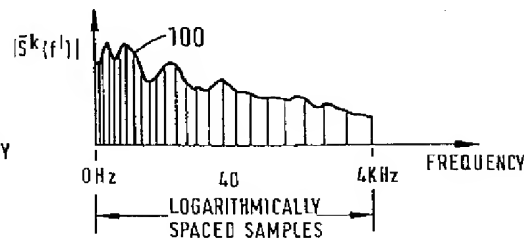


【図20】

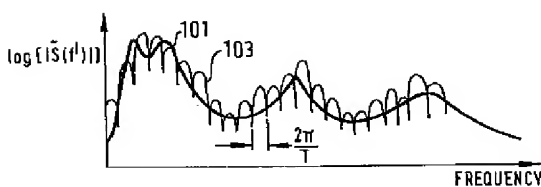
【図6】



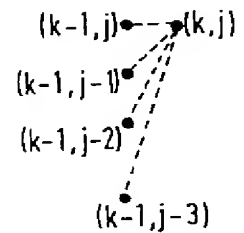
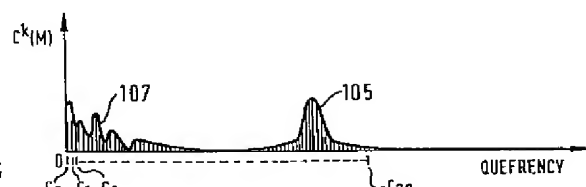
【図7】



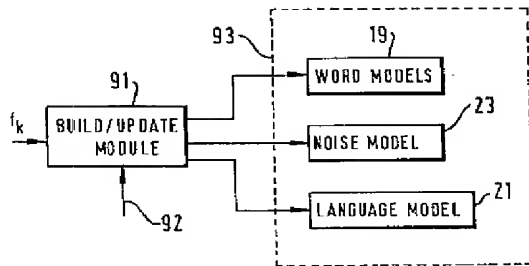
【図8】



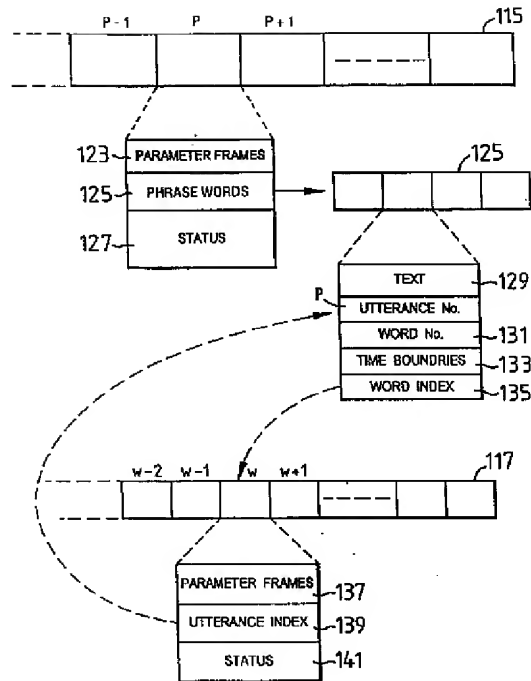
【図9】



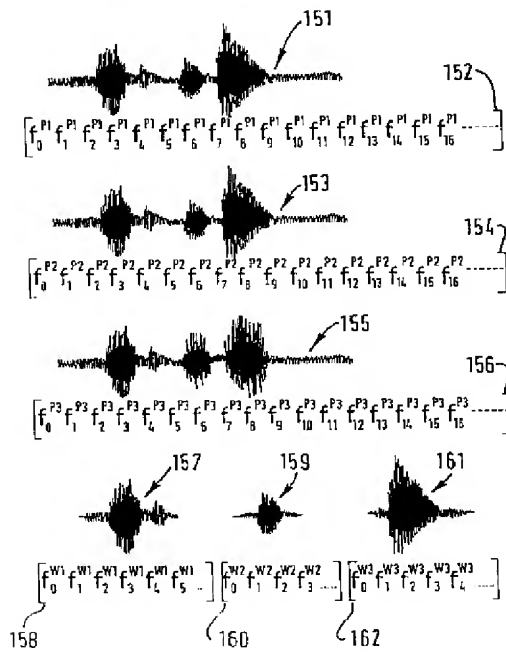
【図10】



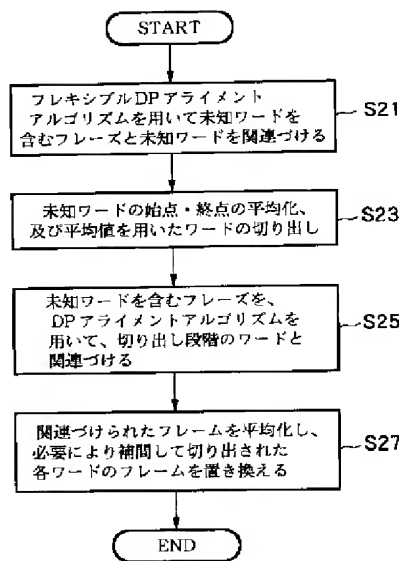
【図12】



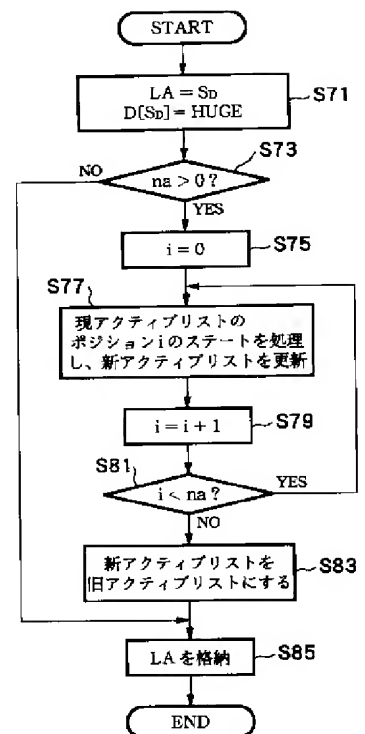
【図13】



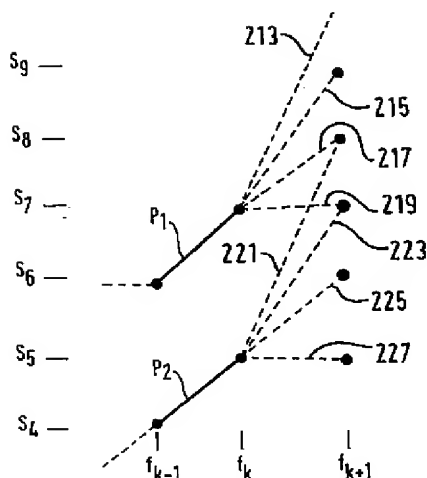
【図14】



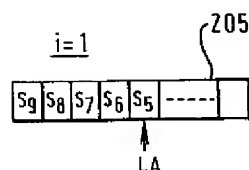
【図24】



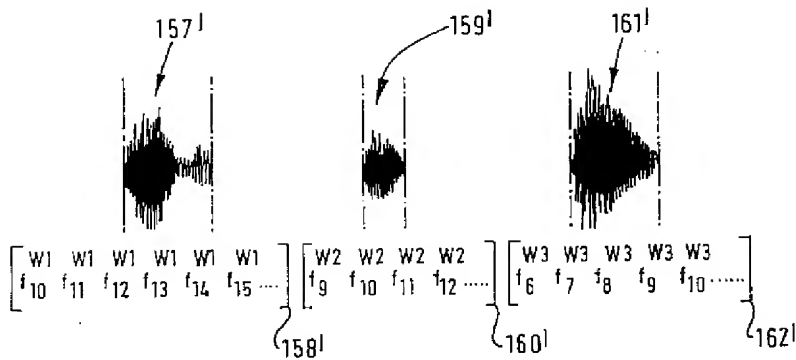
【図25】



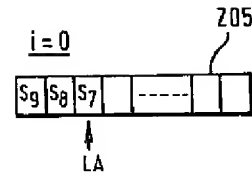
【図26B】



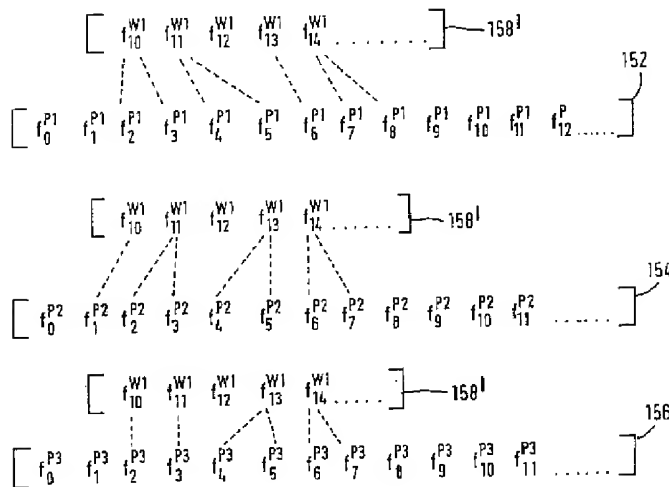
【図15】



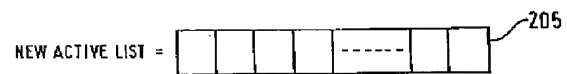
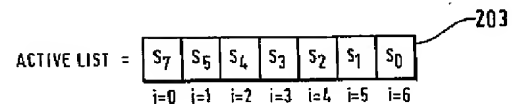
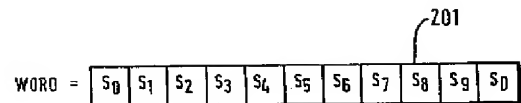
【図26A】



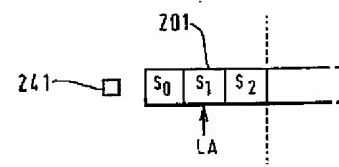
【図16】



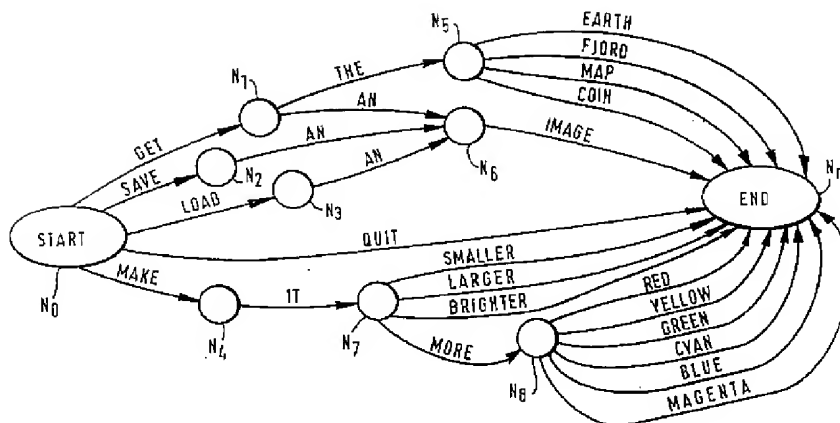
【図22】



【図31】

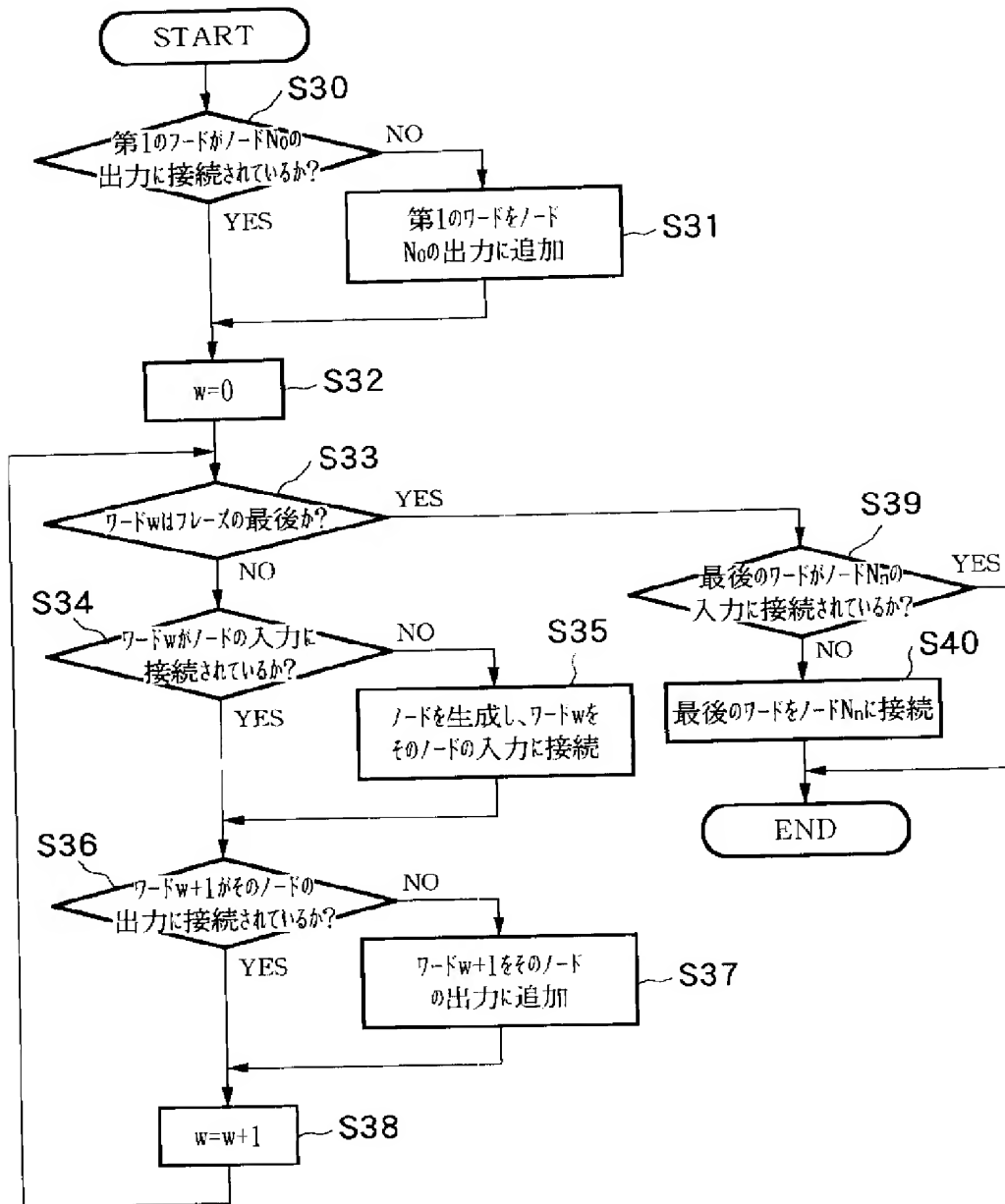


【図17A】

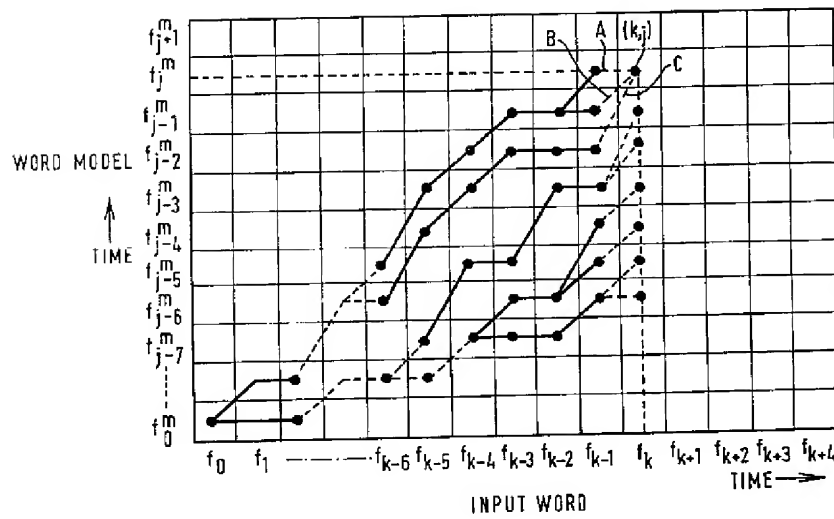




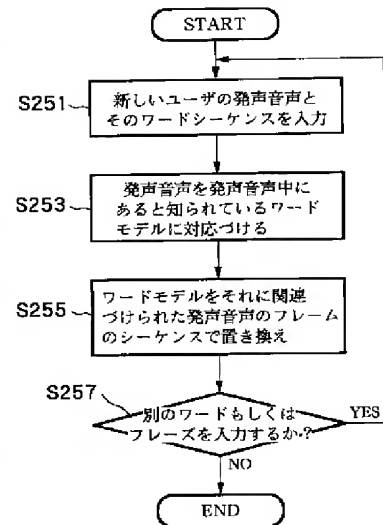
【図17B】



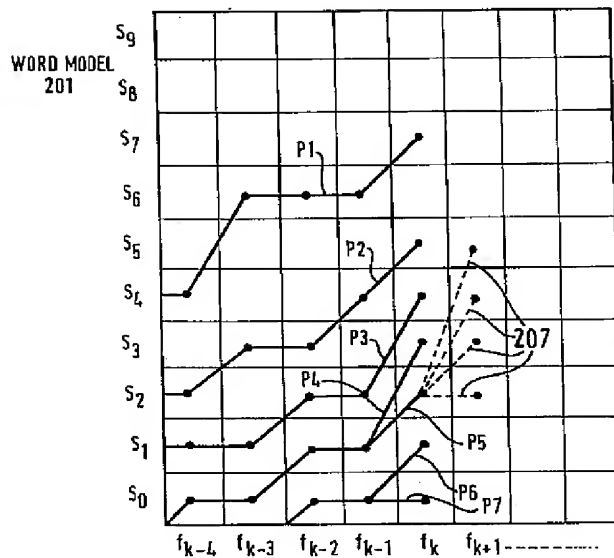
【図18】



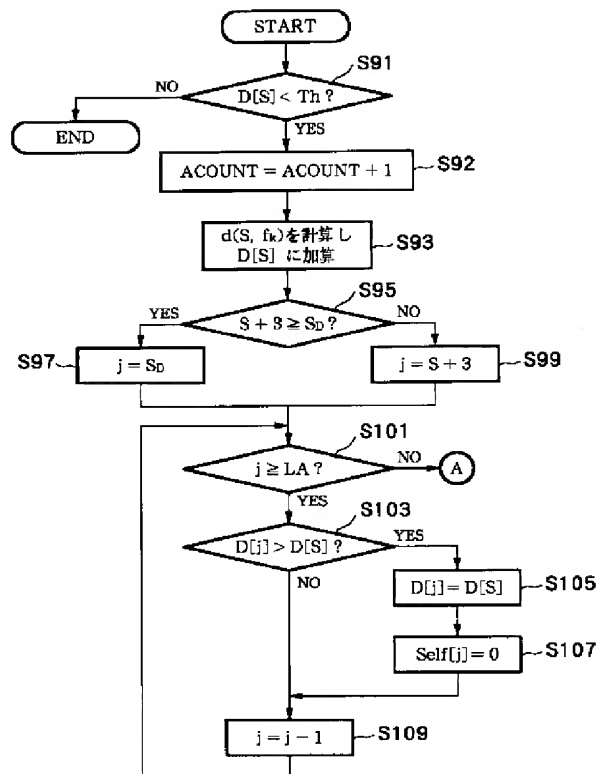
【図34】



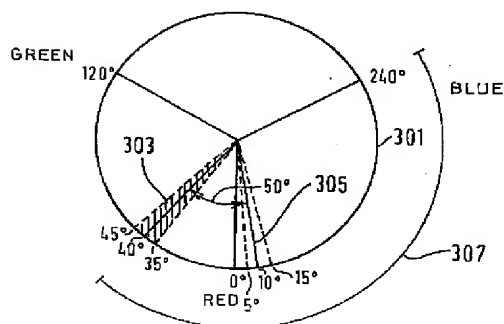
【図23】



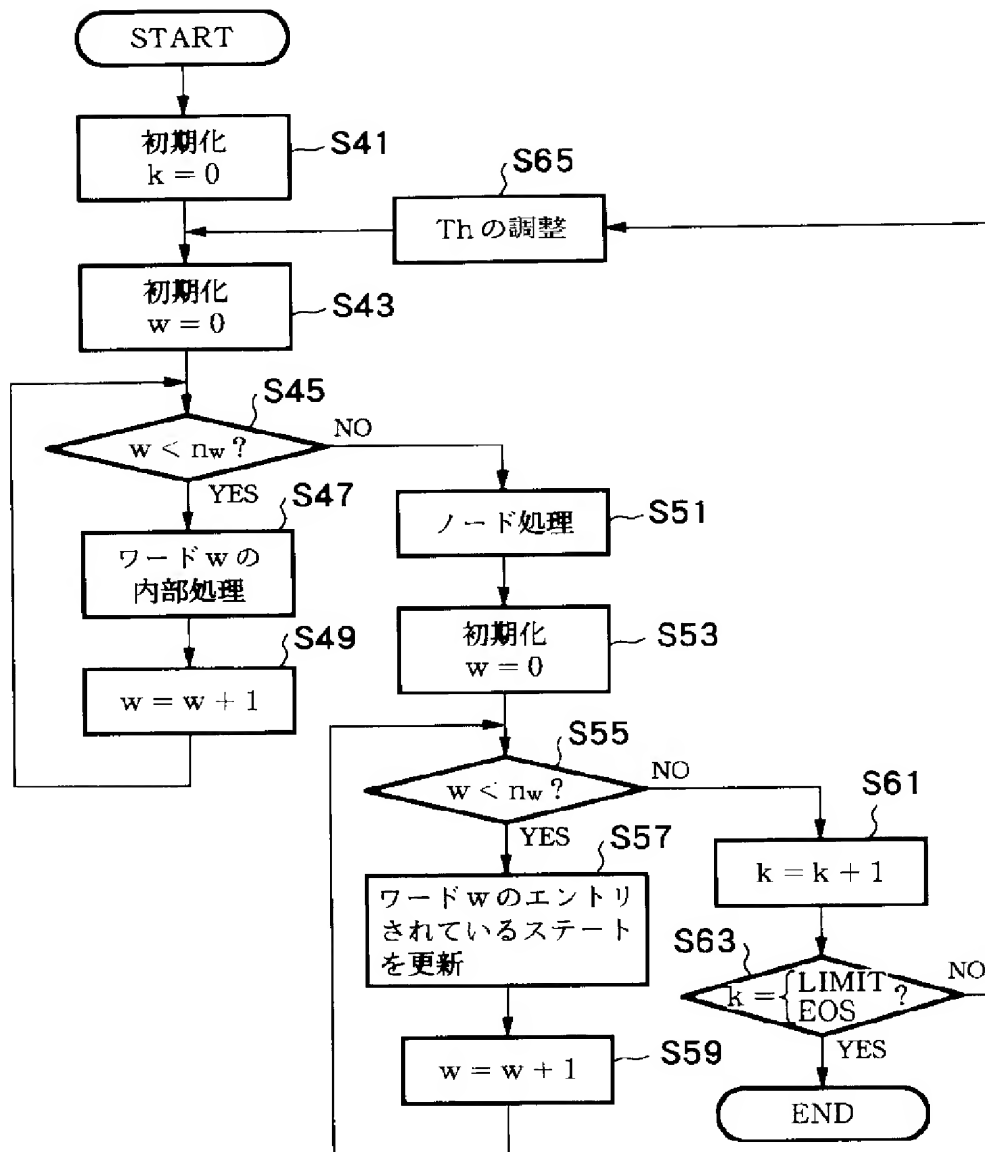
【図27A】



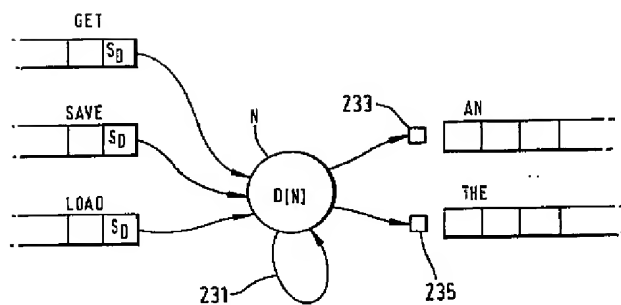
【図38】



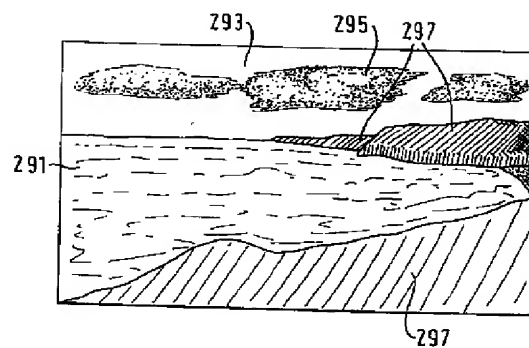
【図21】



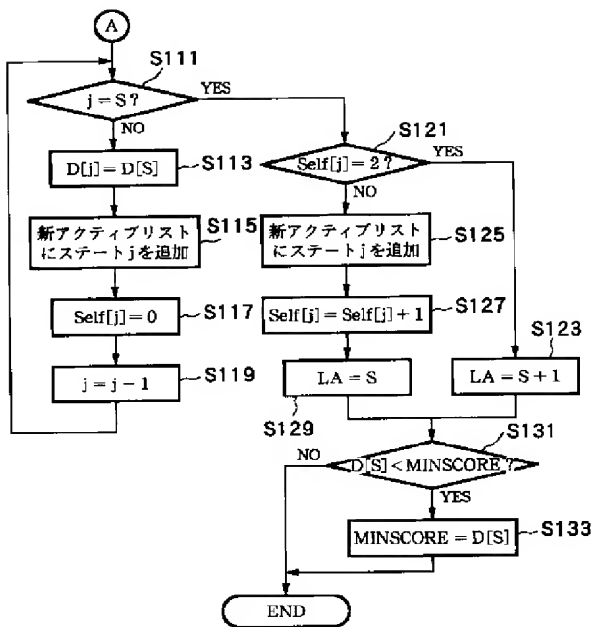
【図29】



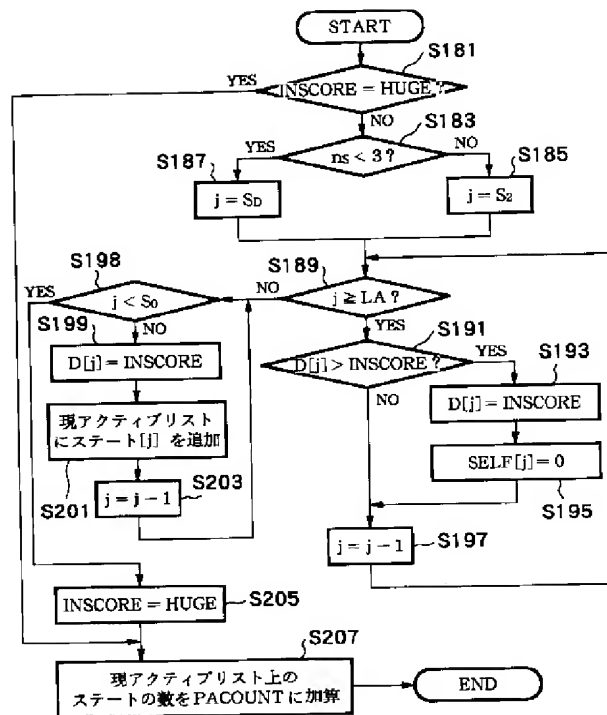
【図37】



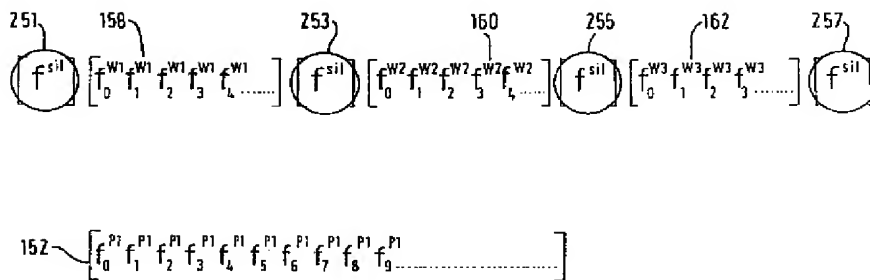
【図27B】



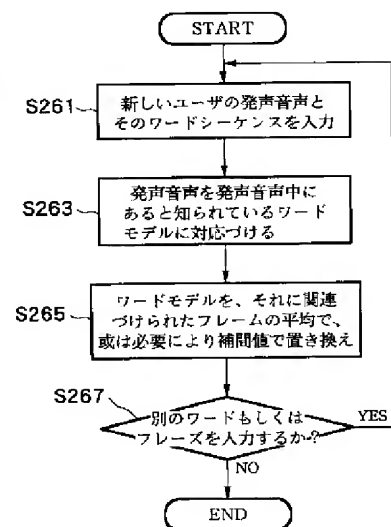
【図30】



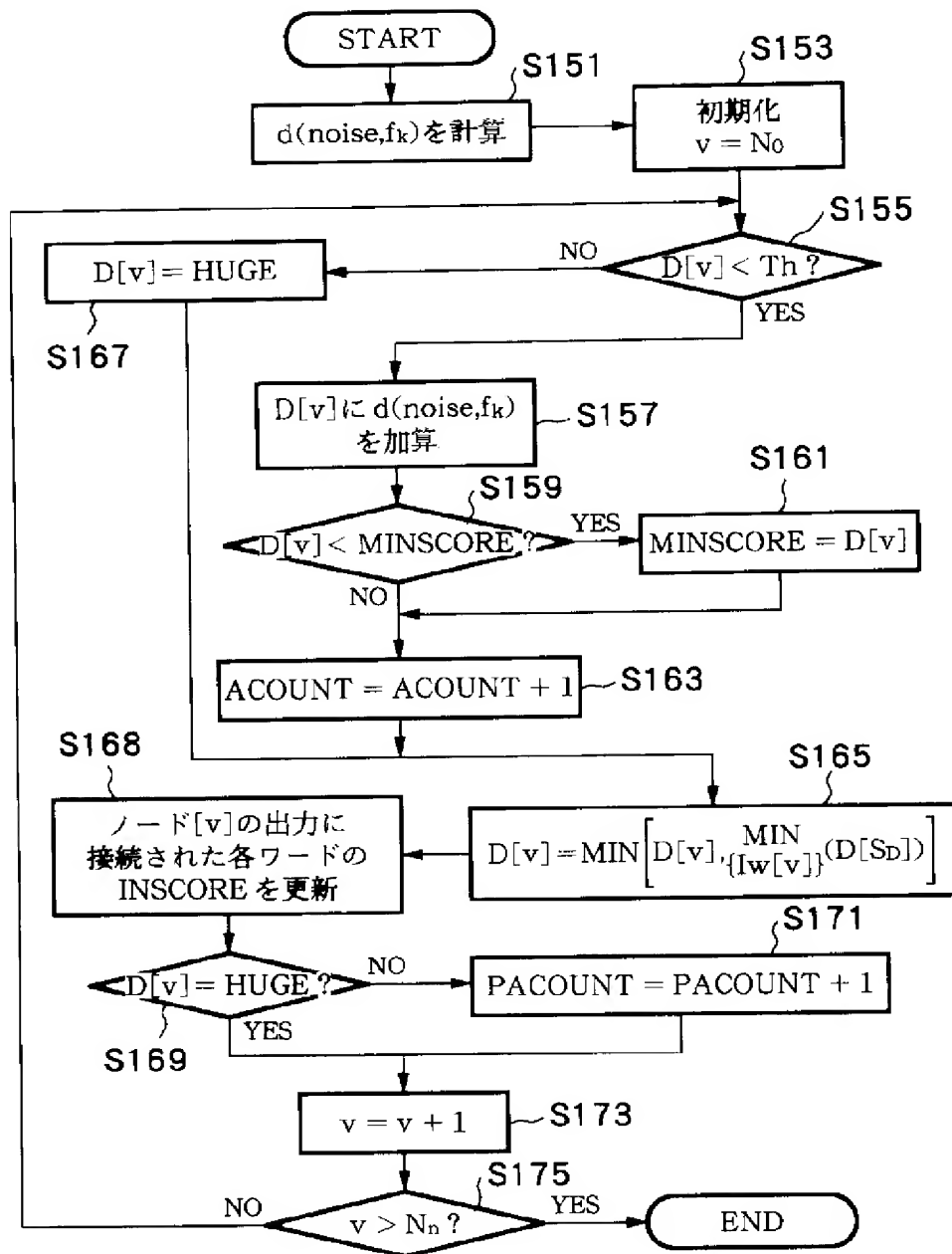
【図33】



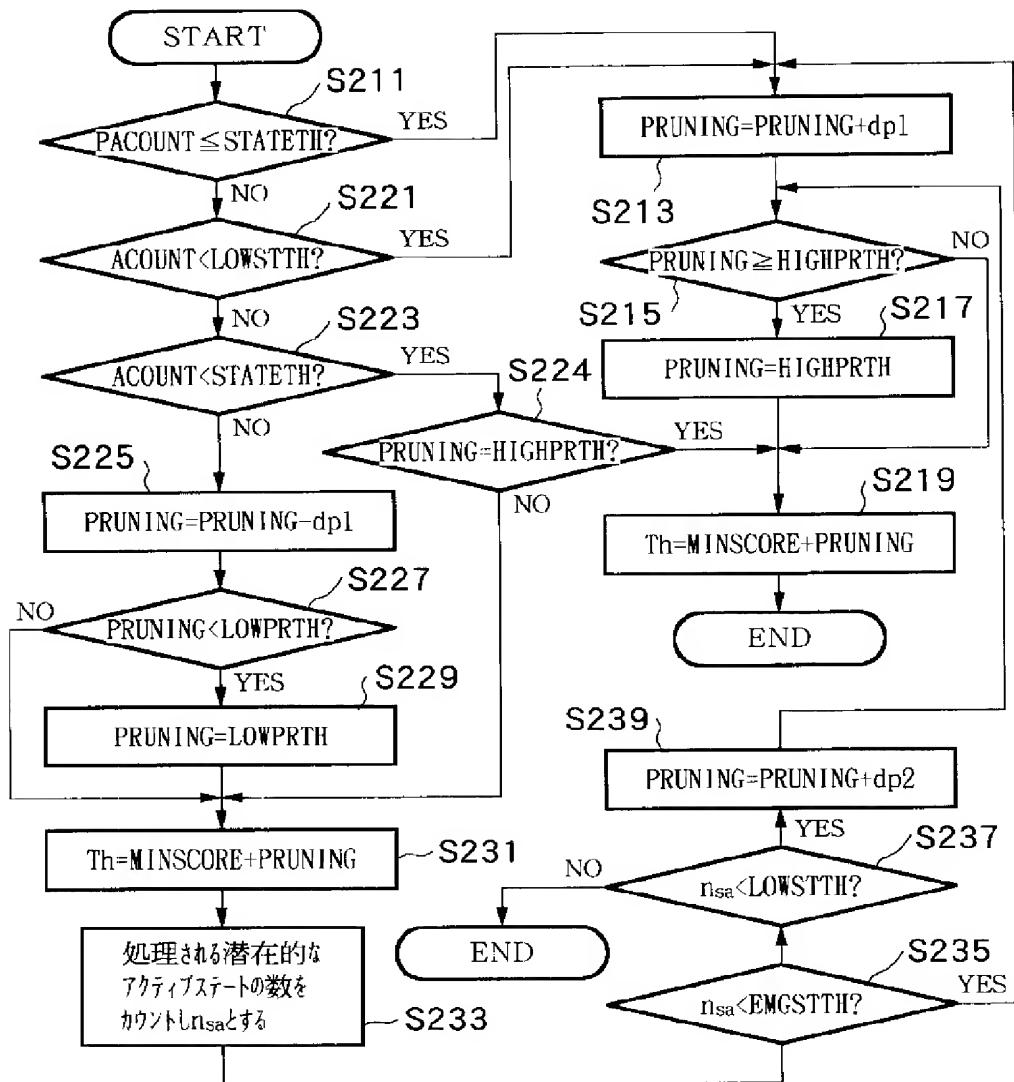
【図35】



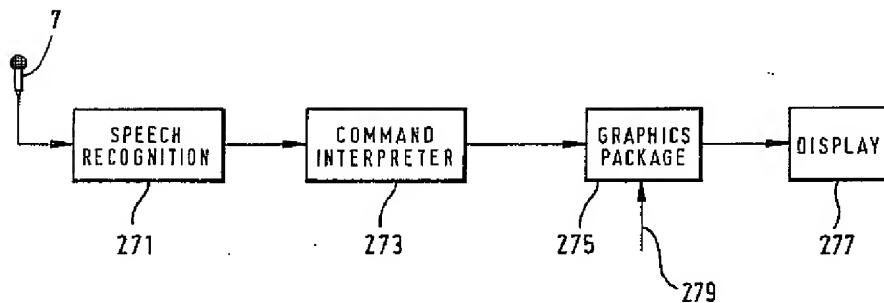
【図28】



【図32】



【図36】



フロントページの続き

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 1 0 L 3/00	5 5 1		G 0 6 F 15/62	3 1 0 A